



National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-56440-7

**A Multi-Level Expert System
for the Segmentation and Interpretation
of Remotely Sensed imagery**

by
Jérôme MIZON

**A thesis submitted to the
School of Graduate Studies and Research
in partial fulfillment of the requirements
for the degree of
Master of Applied Science**

**Ottawa-Carleton Institute for Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
University of Ottawa**

September, 1987



UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

à mes parents

et à mes frères et soeurs.

RÉSUMÉ

L'objectif d'un système de vision par ordinateur est de détecter pour ensuite interpréter les objets présents dans une image. La dernière décennie a vu le développement de tels systèmes utilisant les plus récentes techniques d'intelligence artificielle. Ces systèmes de vision sont en accord avec le paradigme segmentation/interprétation mais restent néanmoins des systèmes qui améliorent une segmentation de bas niveau. De plus, ils n'ont pas été créés pour analyser des images multi-canaux et multi-spectrales bien que les auteurs s'accordent à dire que l'incorporation de telles sources multiples d'informations améliorerait le développement et la qualité des systèmes de vision par ordinateur. Nous avons développé un système de vision par ordinateur utilisant une représentation des connaissances basée sur la notion de prototype (frames) ainsi qu'une structure de contrôle incluant les deux types d'analyse "bottom-up" et "top-down". Ce système est capable de reconnaître certains objets tels que des champs dans des images satellites.

SUMMARY

The objective of a Computer Vision System (CVS) is to outline the objects in a picture and label them with an appropriate interpretation. The last decade has seen the emergence of some CVS's based on the latest techniques of Artificial Intelligence (AI). These systems are built according to the paradigm segmentation/interpretation but are still systems which refine a low level segmentation. Moreover, they are not designed to deal with multi-channel, multi-spectral images. We developed a CVS for the segmentation and analysis of multi-spectral remotely sensed imagery. This system involves a powerful knowledge representation scheme using frames to represent multiple sources of information and a control structure incorporating both bottom-up and top-down analysis to recognize objects such as fields.

ACKNOWLEDGEMENTS

Tout d'abord, je tiens à remercier mon superviseur, Dr M. Goldberg, pour son aide constante et sa confiance lors de la poursuite de ce travail et la rédaction de cette thèse. C'est également grâce à lui que j'ai pu passer ces deux dernières années au Canada.

Je voudrais ensuite remercier mon co-superviseur, Dr D. Goodenough dont les nombreuses idées ont toujours été profitables à ce travail.

Je suis reconnaissant au Centre Canadien de Télédétection de m'avoir permis de travailler avec ses moyens techniques et de m'avoir accueilli de façon très agréable, je pense en particulier à John Zelek, Jean-François Meunier et Gordon Plunkett.

Une note spéciale à Patrick Constant qui m'a initié aux rudiments de l'Intelligence Artificielle de manière amicale et efficace et à Michel Ouellet pour son support moral.

Une note finale pour Jean Sébastien Thériault dont la disponibilité lors de la rédaction de cette thèse fut grandement appréciée.

TABLE OF CONTENTS

| | | |
|---------|---|----|
| 1.0 | Introduction..... | i |
| 1.1 | Artificial Intelligence and Expert Systems: concepts and terminology..... | 1 |
| 1.1.1 | Introduction to Artificial Intelligence..... | 1 |
| 1.1.2 | Introduction to Expert Systems..... | 1 |
| 1.1.3 | The structure of Knowledge Based Systems..... | 4 |
| 1.1.4 | Computer Languages and Tools..... | 6 |
| 1.2 | Computer Vision Systems..... | 8 |
| 1.2.1 | Defining Computer Vision..... | 8 |
| 1.2.2 | Low Level Vision..... | 10 |
| 1.2.3 | High Level Vision..... | 11 |
| 1.2.4 | The Issues in Computer Vision..... | 14 |
| 1.3 | VISIONS, SPAM and SIGMA..... | 17 |
| 1.3.1 | VISIONS..... | 17 |
| 1.3.1.1 | System Overview..... | 17 |
| 1.3.1.2 | Low Level Process..... | 18 |
| 1.3.1.3 | High Level Process..... | 18 |
| 1.3.1.4 | Results and Conclusions..... | 20 |
| 1.3.2 | SPAM..... | 21 |
| 1.3.2.1 | System Overview..... | 22 |
| 1.3.2.2 | Low Level Process..... | 22 |
| 1.3.2.3 | High Level Process..... | 22 |
| 1.3.2.4 | Results and Conclusions..... | 24 |

| | | |
|---------|--|----|
| 1.3.3 | SIGMA..... | 25 |
| 1.3.3.1 | System Overview..... | 25 |
| 1.3.3.2 | Low Level Process..... | 26 |
| 1.3.3.3 | High Level Process..... | 26 |
| 1.3.3.4 | Results and Conclusions..... | 29 |
| 1.3.4 | Overall Conclusion..... | 29 |
| 2.1 | LOUSTIC, a Frame Based Computer Vision System..... | 32 |
| 2.2 | System Organization..... | 34 |
| 2.3 | The Low Level Vision System..... | 37 |
| 2.3.1 | The Very Low Level Knowledge Acquisition System..... | 37 |
| 2.3.2 | The Low Level Knowledge Acquisition System..... | 44 |
| 2.4 | The High Level Vision System..... | 46 |
| 2.4.1 | The High Level Knowledge Acquisition System and Bottom-up Interpretation..... | 46 |
| 2.4.2 | Bottom-up Interpretation using the Evidence Accumulation Approach..... | 46 |
| 2.4.3 | Top-down Interpretation..... | 48 |
| 2.5 | Scheduling and control, the construction of interpretation..... | 49 |
| 2.6 | Implementation..... | 75 |
| 2.7 | Results..... | 76 |
| 2.7.1 | Original Image..... | 76 |
| 2.7.2 | Segmented Picture and low level routines..... | 77 |
| 2.7.3 | Experiments..... | 79 |
| 2.8 | Summary and conclusions..... | 85 |
| 3.1 | References..... | 88 |
| | Appendix: a typical example session..... | 92 |

LIST OF FIGURES

| | | | |
|----|-------------|---|----|
| 1 | Fig 1.1.2 | - Knowledge engineering, transferring knowledge from an expert to a computer program..... | 2 |
| 2 | Fig 1.1.4 | - Components of a support environment for expert systems..... | 7 |
| 3 | Fig 1.2.1.1 | - Representation of a Computer Vision System..... | 9 |
| 4 | Fig 1.2.1.2 | - The Computer Vision Paradigm..... | 9 |
| 5 | Fig 1.2.2.1 | - Various Segmentation Techniques..... | 11 |
| 6 | Fig 1.2.3 | - Interpretation: The Strategies..... | 15 |
| 7 | Fig 1.2.4 | - Timelines for Potential Applications of AI in CVS's..... | 16 |
| 8 | Fig 1.3.1.1 | - VISIONS..... | 17 |
| 9 | Fig 1.3.1.2 | - Example sketch of a partial model..... | 19 |
| 10 | Fig 1.3.2.1 | - Overview of SPAM..... | 21 |
| 11 | Fig 1.3.3.1 | - Overview of SIGMA..... | 25 |
| 12 | Fig 1.3.3.3 | - Typical network for an outdoor scene..... | 27 |
| 13 | Fig 2.2.1 | - System organization of LOUSTIC..... | 35 |
| 14 | Fig 2.3.1 | - Very Low Level Knowledge Acquisition System..... | 38 |
| 15 | Fig 2.3.2 | - Spatial attributes of a segment..... | 42 |
| 16 | Fig 2.3.3 | - Segments Linked Intersection of windows..... | 43 |
| 17 | Fig 2.3.4 | - Spatial relations between segments..... | 43 |
| 18 | Fig 2.4.1 | - Interpretation stages: the strategies..... | 48 |
| 19 | Fig 2.4.2.1 | - Spatial links of objects in a satellite picture..... | 47 |
| 20 | Fig 2.5.1 | - Scheduler..... | 49 |
| 21 | Fig 2.5.2 | - Segment_02, Segment_04 and intersecting segments..... | 51 |

| | | | |
|----|-------------|--|----|
| 22 | Fig 2.5.3 | - Example of a split : Conditions..... | 54 |
| 23 | Fig 2.5.4 | - Example of a split : Action..... | 55 |
| 24 | Fig 2.5.5 | - New created segment..... | 53 |
| 25 | Fig 2.5.6 | - Merge of a small segment: Conditions..... | 57 |
| 26 | Fig 2.5.7 | - Merge of a small segment: Action..... | 58 |
| 27 | Fig 2.5.8 | - Impossible merge of a small segment : Conditions..... | 60 |
| 28 | Fig 2.5.9 | - Impossible merge of a small segment : Action..... | 61 |
| 29 | Fig 2.5.10 | - End of the analysis for segment_02..... | 63 |
| 30 | Fig 2.5.11 | - Merge of segment : Conditions..... | 65 |
| 31 | Fig 2.5.12 | - Merge of a segment : Action..... | 66 |
| 32 | Fig 2.5.13 | - End of the analysis for segment_02 and segment_44..... | 67 |
| 33 | Fig 2.5.14 | - Bottom-up interpretation of a field : Conditions..... | 69 |
| 34 | Fig 2.5.15 | - Bottom-up interpretation of a field : Action..... | 70 |
| 35 | Fig 2.5.16 | - Interpretation of a field using evidence accumulation Conditions..... | 72 |
| 36 | Fig 2.5.17 | - Interpretation of a field using evidence accumulation Action..... | 73 |
| 37 | Fig 2.7.1.1 | - Original Landsat image, channels 1,2,3..... | 76 |
| 38 | Fig 2.7.1.2 | - Original Landsat image, channels 1,4,6..... | 77 |
| 39 | Fig 2.7.2 | - Segmented picture..... | 78 |
| 40 | Fig 2.7.3.1 | - Interpreted segments..... | 81 |
| 41 | Fig 2.7.3.2 | - Bottom-up interpretation of a road : Conditions..... | 83 |
| 42 | Fig 2.7.3.3 | - Bottom-up interpretation of a road : Action..... | 84 |

LIST OF TABLES

1 Table 1.2.2.1 - Some questions about segmentation..... 12

LIST OF ACRONYMS

| | |
|---------|---|
| AI | Artificial Intelligence |
| AKO | A Kind Of |
| CCRS | Canada Centre for Remote Sensing |
| CVS | Computer Vision System |
| HLFB | High Level Frame Base |
| HLKAS | High Level Knowledge Acquisition System |
| HLRB | High Level Rule Base |
| HLVS | High Level Vision System |
| ILFB | Intermediate Level Frame Base |
| IUS | Image Understanding System |
| KAS | Knowledge Acquisition System |
| LLKAS | Low Level Knowledge Acquisition System |
| LLVS | Low Level Vision System |
| LTM | Long Term Memory |
| QAM | Query Answering Module |
| RESHELL | Remote Sensing Expert System Shell |
| STM | Short Term Memory |
| VLLFB | Very Low Level Frame Base |
| VLLKAS | Very Low Level Knowledge Acquisition System |

1.0 Introduction

Computer Vision has been a major subfield of Artificial Intelligence during the last twenty years. A Computer Vision System receives as input an image represented as a large number of pixels and provides as output a symbolic description of the scene under analysis. Some problems remain to be solved in order to effectively interpret an image. These problems involve the low level processes which perform a segmentation of the image and extract statistics on these segments, as well as the high level processes which attempt to interpret the image.

Although many algorithms exist to segment an image at the low level process, there is no algorithm which provides the high level system with the best possible segments for later interpretation. Instead, what results are some regions of the segmented picture which exhibit "oversegmentation" and others which result from an "undersegmentation". The problems involved in the high level processes concern specific Artificial Intelligence topics such as the knowledge representation of the low level segments, the representation of the world under analysis and the strategies to apply for interpretation.

After a review of the terminology and main concepts of Artificial Intelligence and Computer Vision, we investigate three Computer Vision Systems which have been proven to be fundamental to the field. We present their approaches to the issues of knowledge representation and control strategies and show that all of them rely on a general purpose low level segmentation process.

Finally, we present a system which makes use of some of the new concepts and ideas from these systems. Based on the low level-high level paradigm, the system first segments a multichannel, multispectral LANDSAT digital image and extracts some statistics on these segments. Every segment is then translated into the format of a frame, a powerful knowledge representation scheme which enables the integration of multiple sources of knowledge. Then, the system uses both low level knowledge on general segmentation and high level knowledge on the world under analysis to refine the initial low level process and detect the presence of fields in the image. The control strategy for the interpretation process incorporates both bottom-up and top-down analysis. Applied on a digital LANDSAT picture of Saskatchewan, the system improved the initial segmentation by merging and splitting segments and was able to detect and interpret 65 % of the fields present in the image according to the knowledge it had on fields.

This system is among the first attempts to design and implement an intelligent system which has to deal with multispectral, multichannel digital images. It provides a set of useful tools for symbolic image analysis as well as different control strategies for the building and testing of computer vision systems.

1.1 Introduction to Artificial Intelligence and Expert Systems

1.1.1 Introduction to Artificial Intelligence

Artificial Intelligence is the study of ideas which enable computers to be intelligent [Winston84]. Thus, the goal of AI scientists had always been to develop computer programs that could in some sense think; that is, solve problems in a way that would be considered intelligent if done by a human [Waterman86]. This explains why the major research topics in AI have been the study of Natural Language, Robotics, Vision. However, the area which has received much attention lately is expert systems [Basden84].

1.1.2 Introduction to Expert Systems

Ideally an expert system is a computer program that solves problems in the same manner as a human expert. Nevertheless the definition of an expert system will vary from author to author. A compilation of the different solutions could give for its definition :
"An expert system is an intelligent computer program that uses knowledge, inference mechanisms and control structures to solve problems on a very narrow domain, problems that are difficult enough to require significant human expertise for their solution and which are not solvable in a conventional computer manner" [Rich 83][Basden84][Mooneyhan83].

The task of building an expert system is called Knowledge Engineering and involves at least two persons:

-A Domain Expert who is the expert in the domain.

-A Knowledge Engineer who translates the Domain Expert's knowledge into a more computer-like form. The result will be the Expert System (Fig 1.1.2).

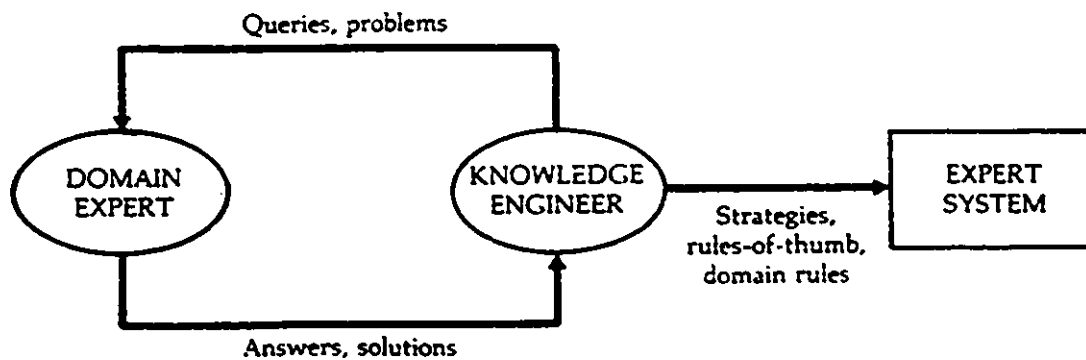


Fig 1.1.2

Knowledge Engineering -
transferring knowledge from an expert
to a computer program.
(from [Waterman86])

Research on expert systems began twenty years ago and since then many (expert) systems have been built. In his book "A Guide to Expert Systems" Waterman identified more than 150 expert systems that have reached the stage of a research prototype in different applications areas such as Agriculture, Chemistry, Engineering, Law, Medicine and Military Science [Waterman86].

Some of the more well known expert systems include:

- **Mycin** which assists physicians in the selection of appropriate antimicrobial therapy for hospital patients [Shortliffe76].
- **Prospector** which acts as a consultant to aid exploration geologists in their search for ore deposits [Duda79].
- **Hearsay II** - a speech understanding system [Erman et al 80].
- **Visions** - an experimental environment for image interpretation [Hanson and Riseman78].

Many authors who develop complex systems based on AI techniques are reluctant to call the systems expert, because they do not perform at the competence level of a human expert, often due to the complexity of the task. These systems have become known as rule-based systems or knowledge based systems [Plunkett86].

1.1.3 The structure of Knowledge Based Systems

Knowledge based systems are structured in many different ways; however, they all contain the following basic components:

- Knowledge Base,
- Inference Engine,
- Working Memory,
- Human Interaction Capability.

The **Knowledge Base** consists of a set of domain facts and heuristics associated with the problem. Knowledge representation is still a major research topic in AI. The most popular way to representing knowledge is by *production rules* (or condition-action rules or if-then rules). Rules provide a formal way of representing recommendations, directives or strategies (in this case they are also called *meta-rules*). Another representation scheme is *the semantic network* where the knowledge is represented as a network structure. A semantic network consists of points called nodes connected by links or arcs describing the relations between the nodes.

Finally, *Frames* are a special way of representing common concepts and situations. A frame is organized much like a semantic network (in fact a system which uses both a semantic network and frames will be called a *frame based system*) and consists of a collection of attributes called slots. A slot can have a value or can be associated with a procedure which is executed when the information in the slot needs to be computed, changed or consulted. Frames are used to represent objects. A copy of a frame, with specific values is called an instance and the process of making instances is called instantiation.

The second element of a knowledge based system is the **inference engine**. The inference engine organizes and controls steps taken to solve the problem. Using a rule-based representation scheme, two problem solving paradigms are used to find solutions.

- *Forward chaining* starts at the conditions of the rules and infers all the solutions and possible conclusions.
- *Backward chaining* supposes that a goal is known, then working backwards from the goal, the system searches for the path leading to the goal.

The Working Memory is a global database that keeps track of the problem status, the input data for the particular problem and the relevant history of what has thus been done.

The Human Interaction Capability or man machine interface may be used in three different modes:

1. Getting an answer to a particular problem: user as a client.
2. Increasing the system knowledge : user as a Knowledge Base.
3. Consulting the knowledge base for human use : user as a pupil.

The important factors are that machine interaction must be syntactically and semantically correct for both the human and the expert system.

1.1.4 Computer Languages and Tools

The programming languages used in expert systems are generally either problem oriented languages which have convenient features for performing algebraic manipulations (Fortran, Pascal), or symbol-manipulation languages more suitable for AI applications, the most popular being Prolog and Lisp. However, another level of programming called Knowledge Engineering System or Expert System Shells has been developed. A Knowledge Engineering System is a sophisticated tool for developing expert systems, consisting of an expert system building tool integrated into an extensive support environment.

Early examples of such shells were, in fact, expert systems with their domain specific knowledge removed, leaving the inference engine and support facilities. Thus, **Prospector** became **Kas** (**K**nowledge **A**cquisition **S**ystem), **Mycin** became **Emycin** and **Hearsay I** became **Hearsay III**. Some commercial versions of Knowledge Engineering Systems are also available. The main features of these shells are given in Fig 1.1.4 and good overviews can be found in [Waterman86] and [Gevarter87].

Parallel to the development of Knowledge Based Systems, the field of Computer Vision has received considerable attention from AI labs, for it is one of the main goals of AI scientists to imitate human vision. Some attempts have been made to create and build knowledge-based computer vision systems. After an overview of computer vision systems in Chapter 2, Chapter 3 will review in some details three of these pioneering systems.

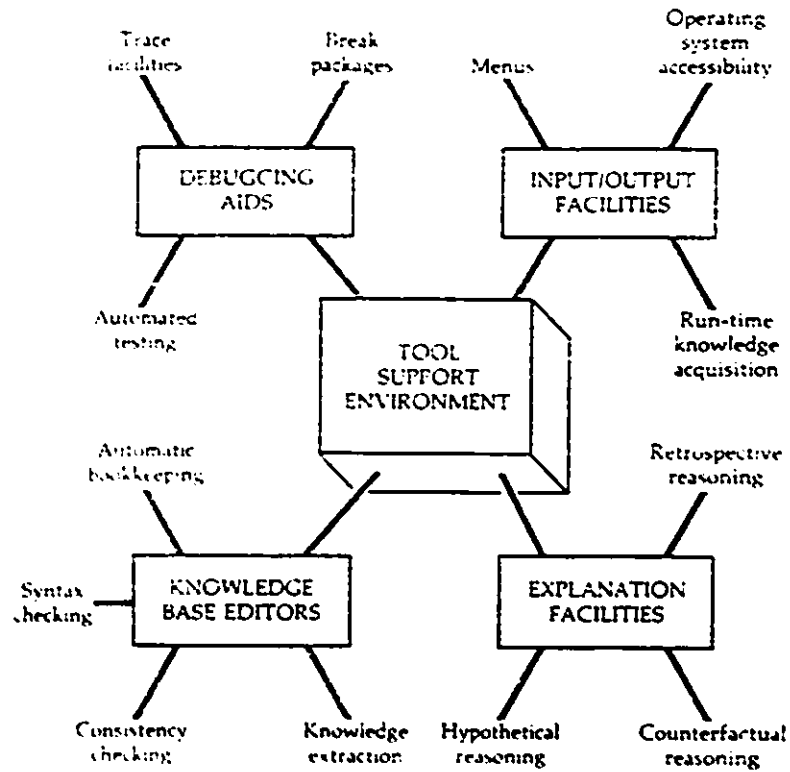


Fig 1.1.4

Components of a support environment

for expert systems tools

from [Waterman86].

1.2 Computer Vision Systems

1.2.1 Defining Computer Vision

Broadly stated, the goal of research in computer vision is the understanding of complex visual processes and the construction of an effective Computer Vision System (CVS), also called an Image Understanding System or an Automatic Target Recognition System. A CVS receives as input a scene in terms of large arrays of digitized sensory data. A typical LANDSAT digital picture, for instance, has a size up to 8000 * 8000 picture elements (called pixels), with up to 7 distinct channels. As an output, the CVS provides a symbolic description of the scene or information relevant to the specific goals of the system (Fig 1.2.1.1).

During its analysis, the CVS will perform at least two types of tasks:

- segmentation
- interpretation

This duality, segmentation-interpretation, is the most commonly used computer vision paradigm. Although recent research tends to incorporate segmentation and interpretation into a single uniform process, the most popular CVS's were developed according to this paradigm [MacKeown85,86],[Hanson and Riseman78],[Matsuyama86,86]. As segmentation is only one step in a series of preliminary processes on the pictures, we will refer to this paradigm as low level vision-high level vision (Fig 1.2.1.2).

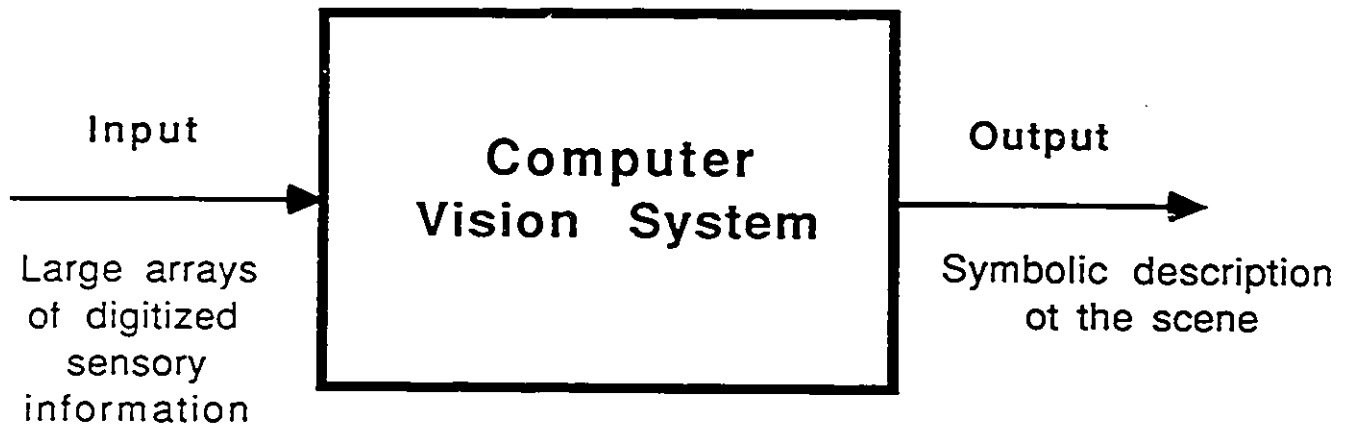


Fig 1.2.1.1

REPRESENTATION OF A COMPUTER VISION SYSTEM

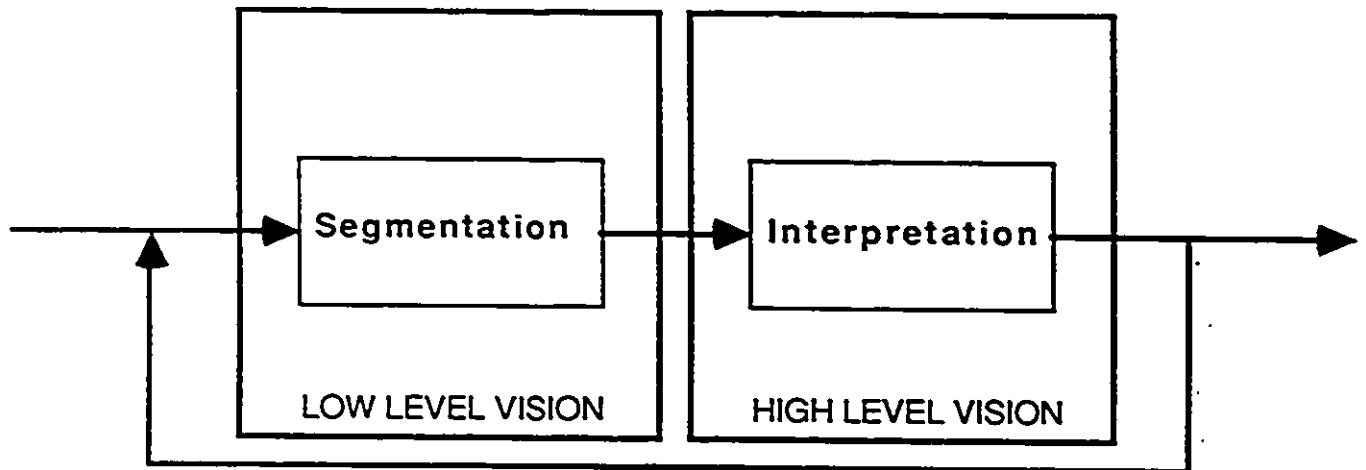


Fig 1.2.1.2

THE COMPUTER VISION PARADIGM

1.2.2 Low Level Vision

The purpose of any Low Level Vision System (LLVS) is to provide the High Level Vision System (HLVS) with image structures that can be associated with objects in the given model. That is, the information from the original picture must be transformed into a more descriptive form, one which captures important visual characteristics of the data and helps to achieve the system's goals [CVS78]. Basic techniques include the segmentation of the image into lines, regions or both lines and regions depending on some criteria.

Many techniques have been (and will be) developed for this segmentation process as it is a major component of a CVS. Fig 1.2.2.1 presents a review of the various techniques used in the CVS community. As there does not appear to exist an optimal segmentation algorithm, many different techniques have evolved in order to provide the HLVS with the most accurate segmentation. Some efforts have been made to integrate semantic processing into the first stages of segmentation [Levine84].

Following this segmentation process, a set of low level routines will normally extract features from the segmented and original pictures. These features include the different spectral values, the shape of a segment, the location of the segment.

Finally, to conclude on low level vision and segmentation, Table 1.2.2.1 presents some questions that arose during one of the first conferences on Computer Vision in 1977 [CVS78]. What is interesting to note, is that 10 years after this conference, the majority of these questions have not yet found a definitive answer and are still current.

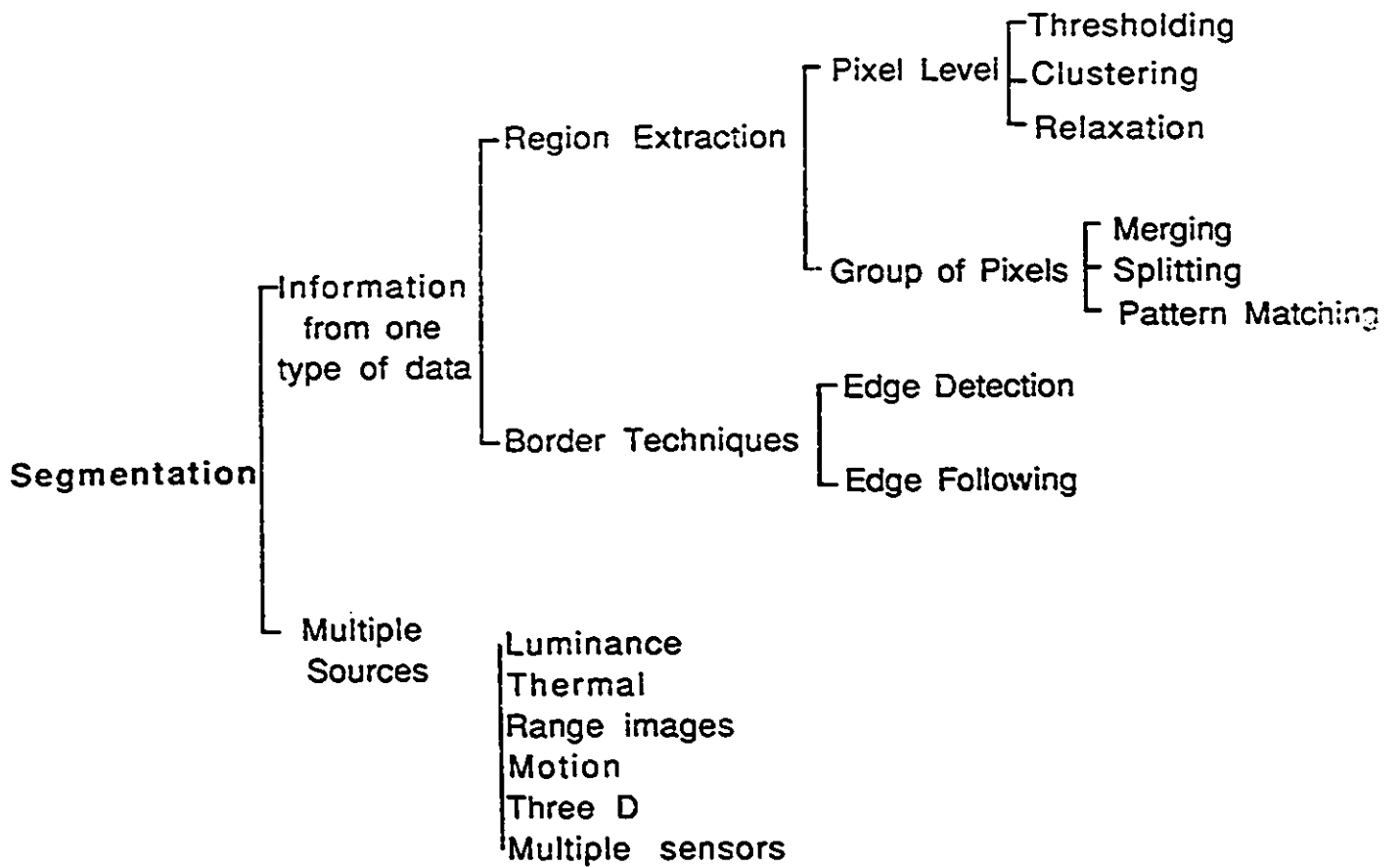


Fig 1.2.2.1
Various Segmentation Techniques

Segmentation

1. What should be the primary basis of segmentation: regions, edges, clusters in feature space? Are these equivalent? (How) should they be combined?
2. How can multiple features and sensory modalities, including color, texture, depth, motion, be used to simplify segmentation? What are the current limitations in reliably processing each of those features? Given these difficulties, what is the relative utility of each to the segmentation process.
3. Which techniques will be powerful enough to deal with the textural variations of natural scenes? What techniques (e.g., histograms) are necessary (or desirable) to extract global feature activity? How can different levels of texture (micro-macro) be detected and utilized?
4. Do parallel vs. sequential approaches lead to different methodologies? To what extent should we be concerned with the development of parallelism at this point?
5. How does the resolution of the sensory data affect the segmentation process and ultimate performance of the system? Can crude segmentation be obtained from coarsely sampled data? Should we be considering the equivalent of eye movements and/or foveation? Is the processing of a hierarchy of resolution levels helpful? How can a scene be 'glanced at' to quickly determine and locate one or two objects or relationships of immediate interest?
6. How can (or should) semantic knowledge be used to guide low level segmentation? What level(s) of knowledge is needed e.g., domain independent interpretation of grey levels as surfaces; domain specific interpretation of regions as objects?
7. How well can (must) segmentation be performed at the retinal level without explicit knowledge? With only domain independent knowledge?

tional Questions Raised During January, 1977

1. Use of Knowledge at the Segmentation Level: The majority of those addressing this question took the position that general vision systems will require at least a first segmentation phase which is knowledge independent. On the other hand, fairly effective results using domain dependent knowledge have been cited. Can effective general segmentation results be obtained without semantics? In order to do so, several people have argued that powerful data grouping and aggregation procedures are required but still remain to be developed. Alternatively, if additional knowledge is required, in what manner can it be employed without losing generality and computational feasibility?
2. Processes Must Deal with Partial Hypotheses and Incomplete Segmentation Results: Several contributors felt that a 'perfect' final segmentation is unrealizable; if this is true, then the semantic interpretation of an image must be based on unreliable and incomplete information. Thus errors in choosing among uncertain alternatives are probable and the search space is likely to be large. How can the process(es) for the construction of models be controlled? This leads to questions on backtracking, distributed application of constraints, top-down analyses via frames, schema, or templates, etc. While there have been some successes at the segmentation level, only limited results (from a smaller number of efforts) at the interpretation level have been obtained. The computational requirements for computer vision are enormous and they will not be eliminated simply by focussing on control structures. This leads to questions concerning the adequacy of current methodology and serial computers for research in computer vision and the final hardware structure for practical, working systems.

Table 1.2.2.1

Questions about Segmentation

(from [CVS78])

1.2.3 High Level Vision

The high level vision or interpretation process is the task of constructing mappings between image structures and objects. Two approaches can be distinguished (Fig 1.2.3) :

- **The bottom-up approach:**

After the image structures have been extracted from the image, these structures are interpreted as instances of the objects in the model. For example , when a large rectangular region is extracted in an outdoor scene picture, it may be interpreted as a house. This strategy is completely data driven.

Recently a new kind of bottom-up strategy was introduced by Matsuyama [Matsuyama86] in his CVS SIGMA : the evidence accumulation approach. This approach will be presented in more detail in the next chapter. The principle is based on the following proposition:

If a structure of type x is present in the scene having certain spatial properties, then there should exist a structure of type y having certain properties in the image.

For example, near a house, one should normally find a road or a driveway. That is, if the system, in analyzing an hypothesis for a house, finds an instance of a driveway near by, then the hypothesis for the house is verified.

- **The top-down approach:**

This approach will first select the appearance of an object and try to find this object with the help of the associated images structures. For example, suppose a CVS wants to find the instance of a house, the CVS will invoke the house model and try to establish the descriptions of the specific image structures to be extracted. This approach is goal directed.

Although it is widely accepted that CVS should incorporate both bottom-up and top-down control processes, the present-day systems generally use only one type of analysis. The main reason is that, until recently, it has not been clear how to integrate both kinds of analysis into an uniform process. One of the first attempts was carried out by Matsuyama in SIGMA [Matsuyama86,86].

1.2.4 The Issues in Computer Vision Systems

Fig 1.2.4 presents the timelines for potential applications of AI in CVS according to Gilmore [Gilmore84]. A certain number of systems have reached the stage of research prototypes and over the past two decades the field has developed from the seminal work of a small number of scientists into a major subfield of AI. The next chapter will present in more details three CVS's:

- VISIONS [Hanson and Riseman78]
- SPAMS [MacKeown85,86]
- SIGMA [Matsuyama86]

VISIONS is more an environment for CVS than a CVS in itself. It was chosen because it introduced important concepts to the Computer Vision community. SPAMS was chosen because it does not restrict its knowledge only to pictures, but also incorporates the knowledge of other sources such as maps. SIGMA is important because its main goal is to incorporate both-top down and bottom -up approaches into a single uniform process.

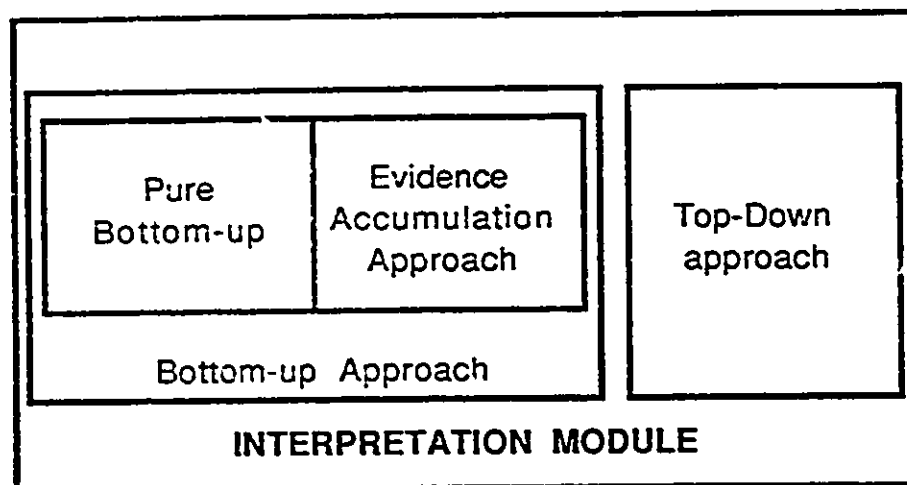


Fig 1.2.3

INTERPRETATION : THE STRATEGIES

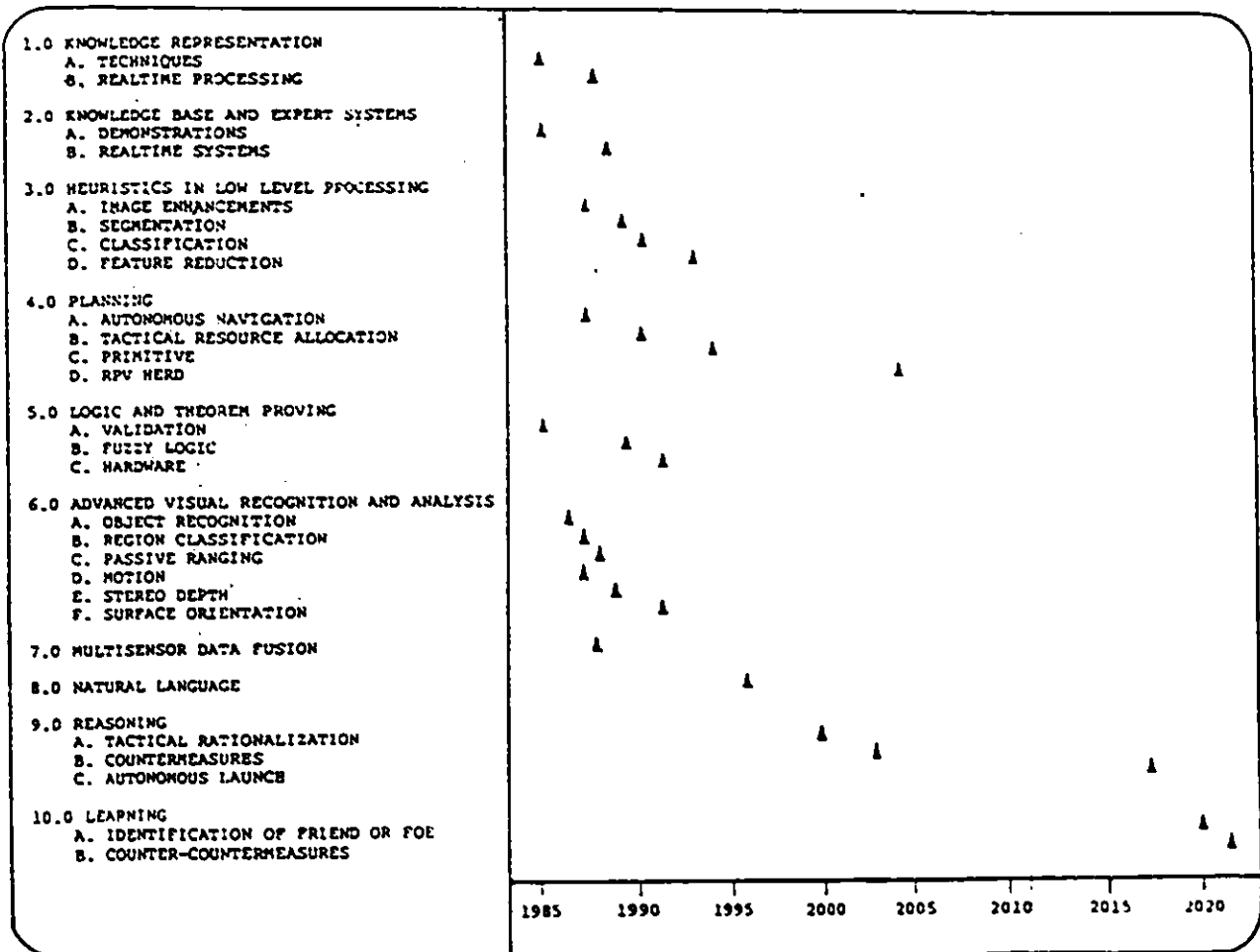


Fig 1.2.4

Timelines for potential applications of AI in CVS's

(from [Gilmore84])

1.3 VISIONS, SPAM AND SIGMA

1.3.1 VISIONS

The goal of VISIONS is the transformation of patterns of sensory visual input data from a bidimensional image of an outdoor scene into a description which captures the meaning of the scene [Hanson and Riseman78]

1.3.1.1 System Overview

As seen in Fig 1.3.1.1, the system is decomposed into two processes: the low level process which operates on numeric arrays of visual data, and the high level interpretation for constructing a description of the scene.

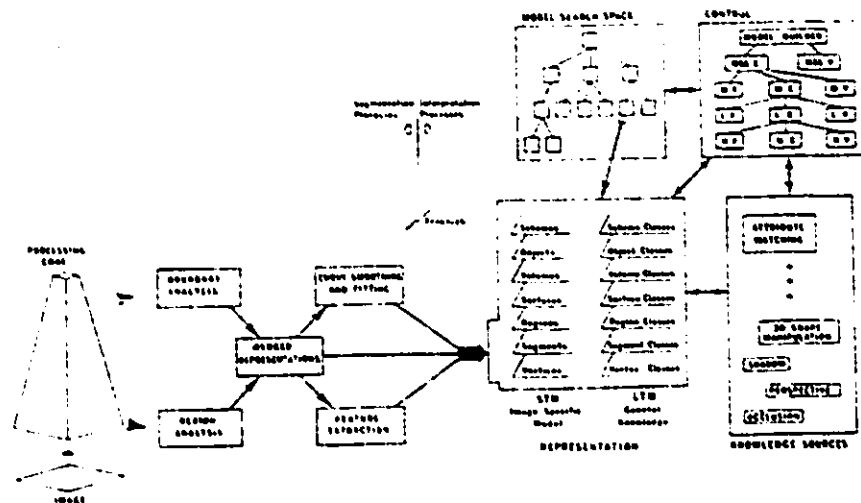


Figure 1. Overview of the VISIONS system. The left hand side represents the segmentation processes which are described in the companion paper in this volume. The right hand side represents the mechanisms for the formation of a descriptive model of a scene. The interpretation components are structurally divided into a representation of knowledge, modular knowledge sources (processes) for the formation of hypotheses, a hierarchical modular strategy to control the application of the knowledge sources, and a tree for representing the history of search through the space of models.

Fig 1.3.1.1

VISIONS

(from [Hanson and Riseman78])

1.3.1.2 Low Level Process

Two distinct algorithms are employed for the segmentation process :

- One algorithm to aggregate edges into boundaries (border technique in Fig 1.2.2.1),
- One algorithm to aggregate pixels into regions (region extraction in Fig 1.2.2.1).

Regions and boundary segments are then smoothed and fitted with simple shapes where possible. Additional features are extracted and associated with each element. The result of this low level process is a layered, directed graph of regions, segments and vertices [Hanson78].

1.3.1.3 High Level Process

• **Knowledge representation:**

VISIONS use two types of memories:

- The Long Term Memory (LTM) which stores general a priori knowledge of the world under analysis.
- The Short Term Memory (STM) which constitutes the system's internal knowledge of the world related to a particular image.

VISIONS also uses the notion of *schemas*, analogous to the notion of frames as described in Chapter 2. These schemas describe prototypical structures (e.g. road, house) .

• Control strategies:

The interpretation of an image is achieved by constructing a description in the STM

(Fig 1.3.1.2). As the paths available for hypothesis formation involve both procedural and declarative knowledge, VISIONS uses both bottom-up and top-down analysis.

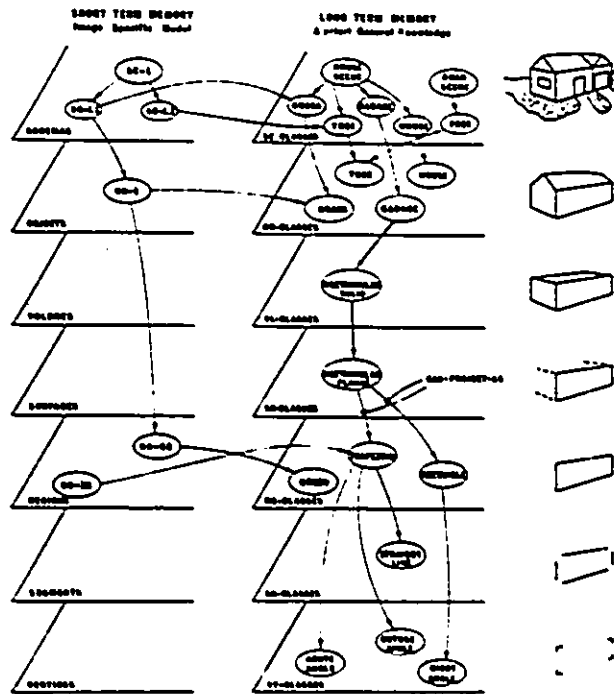


Figure 2. An example sketch of a partial model and the decomposition of declarative knowledge. The knowledge is divided into: 1) a hierarchy of levels of representation defining the key levels of abstraction which are necessary for a general system of visual perception; and 2) short-term memory (STM) representing an interpretation of the specific image and long-term memory (LTM) representing general visual knowledge of the world.

Fig 1.3.1.2

Example sketch of a partial model

(from [Hanson and Riseman78])

1.3.1.4 Results and Conclusions

The main ideas of VISIONS in 1977 include :

- data decomposition into levels, each of which is treated separately,
- independent processes,
- flexible strategies and control processes.

Using this approach, the author believed that the system could be refigured incrementally with modest effort. However, the experiments showed some areas where modifications were necessary. These modifications include :

- The implementation of a feedback from the high level to the low level to redirect segmentation.
- The implementation of other control strategies including top-down, schema-driven control.

1.3.2 SPAM

SPAM is a rule based image interpretation system for the analysis and interpretation of aerial photographs. It coordinates and controls image segmentation analysis and the construction of a scene model. It also provides unique capabilities to bring map knowledge and collateral information to bear during all phases of the interpretation [MacKeown85,86].

1.3.2.1 System Overview

Three main components form the system, as seen in Fig 1.3.2.1:

- an image map database,
- image processing tools,
- a rule based system.

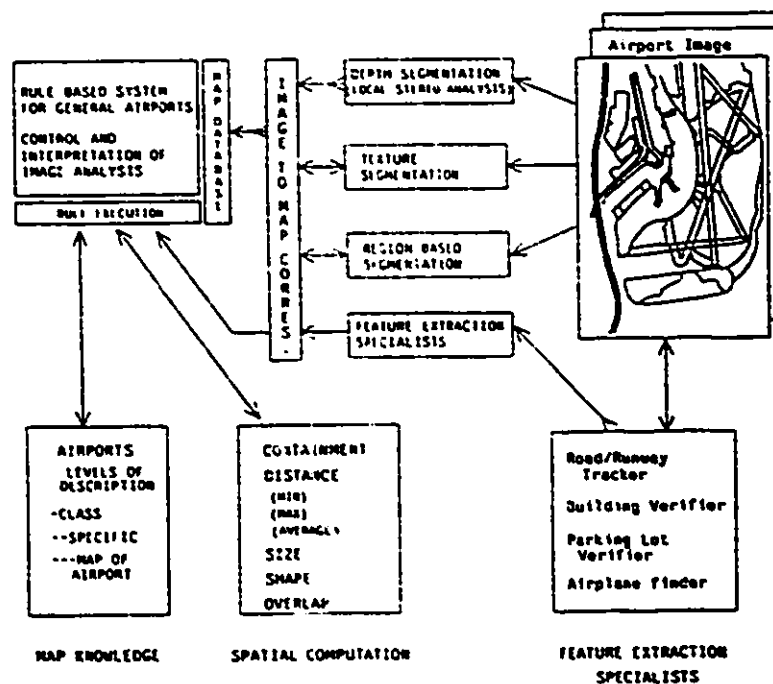


Fig 1.3.2.1

Overview of SPAM

(from [MacKeown85])

1.3.2.2 Low Level Process

The image map database stores facts about the existence and location of man-made natural features. It also provides facilities to compute geometric properties and relationships of map database features.

The image processing tools : The segmentation process uses a region-growing (Fig 1.2.2.1) algorithm, a road / road-like feature follower (border technique) and an interactive, human segmentation system.

1.3.2.3 High Level Process

- **Knowledge representation:**

Four simple working elements are manipulated by the rule-based system:

- **Regions :** Low level image properties of each segmentation region.
- **Fragment:** a fragment is an interpretation for an image region. It contains class and subclass information as well as interpretation status information.
- **Functional areas:** are a collection of man-made and natural features such as routes, runways...
- **Model:** a model is a consistent collection of functional areas.

- **Control system:**

Five processing phases lead to the interpretation:

- **Build** : builds the initial interpretation of region data
- **Local evaluation** : improved processing on the "built" region, generates fragments.
- **Consistency**: computes some statistics and confidence measures on fragments.
- **Functional area** : evaluates fragments that have specific subclass interpretation.
- **Model evaluation** : Fires when sufficient functional areas have been generated.

The strategy is to move sequentially through the five phases. Thus, the picture is interpreted in a bottom-up way.

- **Rule base:**

There are 450 rules grouped into six classes which normally correspond to the interpretation phases.

- Initialization rules.
- Region to interpretation rules.
- Local evaluation rules.
- Consistency rules.
- Functional areas rules.
- Goal generation rules and model selection rules.

1.3.2.4 Results and Conclusions

Implemented in OPS5, SPAM took 9 hours to extract and identify some runways, taxiways and grassy areas for several images of the National airport of Washington D.C. As the author explains, SPAM represents research in progress, and the areas where some problems remain are:

- Reliable low level features from the imagery.
- Design and implementation of effective recognition strategies using the rule based approach.

Further improvements involve:

- The integration of various features such as texture, depth...
- The integration of top-down research with adequate image analysis tools in promising portions of the scene.
- Intensify studies on airport model information.
- Apply SPAM to other airport scenes.

1.3.3 SIGMA

The goal of SIGMA is to develop a robust control strategy for constructing an Image Understanding System (IUS), thus eliminating the need to use large amounts of domain specific control knowledge. SIGMA presents a general framework which enables IUS to integrate both bottom-up and top-down analysis into a single flexible reasoning process. [Matsuyama86,86]

1.3.3.1 System Overview:

SIGMA consists of three main modules (Fig 1.3.3.1) :

- the Low Level Vision System (LLVS),
- the High Level Vision System (HLVS),
- a Query Answering Module (QAM).

The user provides object models to SIGMA and the results of the analysis are available to the user through the Q.A.M.

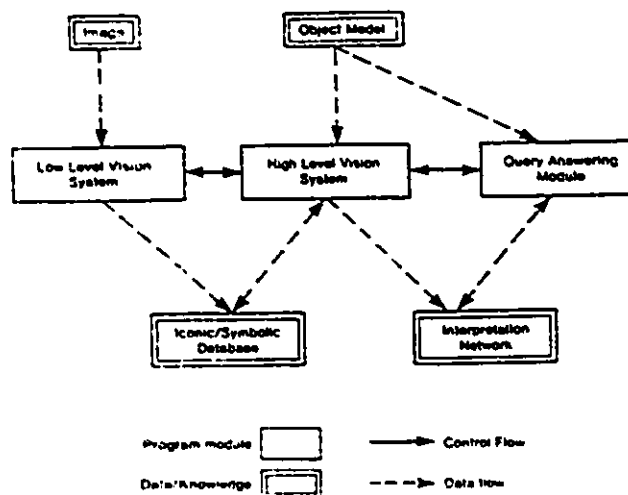


Fig 1.3.3.1

Overview of SIGMA

(from [Matsuyama86]).

1.3.3.2 Low Level Process

The low level vision system of SIGMA has been designed to be a domain-independent, goal-directed segmentation system. It is based on a select-and-schedule strategy, in this way it differs from the two previous approaches described in VISIONS and SPAM. Both VISIONS and SPAM start with a general purpose segmentation algorithm based on regions or lines. The LLVS in SIGMA will basically try to verify some hypotheses generated by the HLVS by selecting the methods which match the hypothesis. Suppose, for example the LLVS is asked to extract all regions which might correspond to house groups and roads in the images. SIGMA then searches for the most promising geometric appearance for houses and roads and applies the best segmentation algorithm it knows to extract this feature (elongated rectangle in this example). The LLVS has also been designed to allow the augmentation or the replacement of low level methods as the algorithms improve.

1.3.3.3 High Level Process

• Knowledge representation:

Basic entities of the representation in SIGMA are frames and are used to model abstract objects in the problem domain such as House or Road. With frames are associated slots where values such as centroid and shapes are recorded. The frames are the nodes in a graph structure where the arcs represent the relations between the objects. These arcs are described with rules or links. The links such as AKO CAN-BE are used to describe the hierarchical relations between objects.

The rules are procedural and they are composed of 3 parts:

<control condition>

<hypothesis>

<action>

Links and/or rules are also stored as slots of the frames. The whole graph represents the object model and a typical graph for the outdoor scene can be seen in Fig 1.3.3.3.

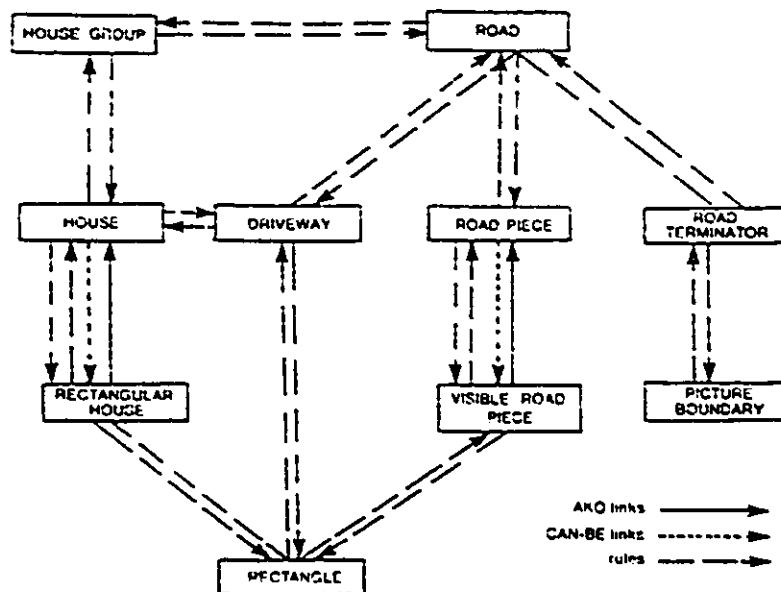


Fig 1.3.3.3

Typical network

for an outdoor scene

(from [Matsuyama86]).

- **Control system:**

The HLVS constructs a partial interpretation based on the result of the initial segmentation. However, during the construction, the HLVS may direct the LLVS to compute more image structures. The HLVS first establishes the local environment, then either bottom-up or top-down processes are activated depending on the nature of this local environment.

- *Bottom-up scheme:* Consider a binary relation $Rel(O_1, O_2)$ between two classes of objects O_1 and O_2 . This relation can be used as a constraint to recognize objects from these two classes. Image structures which satisfy the specified appearances of O_1 and O_2 are extracted and then the system checks that the relation is satisfied by these candidate objects. This particular bottom-up recognition scheme is also called evidence accumulation.

- *Top-down scheme:* Here image structures missed by the initial segmentation are found. The appearance of the object is first determined, and the associated image structures are extracted. For example, suppose the IUS wants to find a house, The IUS invokes the house model and establishes the descriptions of the specific image structures to be extracted from the image.

Five steps are iterated by the HLVS to construct an interpretation:

1. Hypothesis generation
2. Focus of attention
3. Composite hypothesis construction
4. Solution generation
5. Action scheduling.

1.3.3.4 Results and Conclusions

SIGMA has been implemented in Flavor, LISP and C on a VAX 780. It is able to interpret simple black and white aerial photographs by finding houses and parts of roads.

The problems to be solved according to the author are:

- introduction of negative sources of evidence,
- coordination of interpretations executed in parallel in local areas.

1.3.4 Overall Conclusions

As we have seen so far, CVS's consist of two main modules: a low level vision module for segmentation and a high level vision module for interpretation. As Matsuyama says, a powerful vision system should rely on a balance of performance between these two types of modules. The low level module should provide descriptive information about the image to the high level module and the high level module should provide hints about image structure to the low level module [Matsuyama86]. Of the three systems reviewed, SIGMA is the only one which attempts to incorporate bottom-up and top-down analysis into a single uniform process. Both SPAM and VISIONS pay lip-service to the idea but still use a general purpose segmentation algorithm for the low level process. They are not the only ones, as the majority of computer vision systems conclude in this way.

On the other hand, all three systems present the problem of more specific control for the interpretation :

- Top-down scheme for VISIONS and SPAM.
- General control scheme for SIGMA.

For the knowledge representation, each system has its own specific way of representing knowledge. There exists some convergence in the ideas (functional areas of SPAM and iconic description of SIGMA, schemas of VISIONS and frames of SIGMA) and no major divergence of use. All three systems rely on modularity as the best way to enlarge knowledge and control source. The notion of frames appears as a flexible and versatile way to represent different kinds of knowledge:

- pictorial: spectral values and low level coefficients,
- relational: use of inheritance,
- procedural: a slot may be a rule.

The future of Computer Vision seems to be toward modular, rule-based, knowledge-based systems incorporating multiple sources of knowledge such as pictorial data and cartography data, and where segmentation and interpretation are performed at the same time using both bottom-up and top-down control schemes.

We have to emphasize that the current CVS's are still systems that refine a low level segmentation and then try to match the new refined segments with objects present in the scene according to some high level strategies. One of the first attempts to incorporate knowledge on segmentations in a knowledge-based system is due to Levine et al [Levine84]. The primary objective of their system was to organize and control knowledge on segmentation algorithms based on border techniques and region analysis in order to provide a future high level interpretation system with areas of interests.

The next part of this work presents a similar system for the segmentation and analysis of satellite images, LOUSTIC. Only a few algorithms exist for the segmentation of multichannel, multispectral images although agreement is general that the use of such multiple types of data should increase the development process of CVS's [Mooneyhan83]. Based on an initial segmentation algorithm, LOUSTIC uses knowledge about segmentation to improve the quality of the segmented picture. Then, using knowledge about the nature of objects present in the image, it tries to detect and interpret some of these objects. LOUSTIC also attempts to incorporate both bottom-up and top-down analysis for the interpretation stage and uses a knowledge representation scheme based on frames to represent the segments.

2.1 LOUSTIC, a Frame Based Computer Vision System

LOUSTIC is an interactive frame based system for the segmentation and the interpretation of agricultural fields in LANDSAT images. The main goal of this research is to prove that using human knowledge and an adequate and robust control strategy, the building of a knowledge based Computer Vision System recognizing some features in satellite pictures is possible. Other important ideas are:

- The use of a segmentation algorithm for the segmentation of multispectral and multichannel images. None of the systems reviewed in this work and none of the best known CVS are designed to deal with multichannel, multispectral digital images, although all of the authors agree that the use of this type of data should increase and perhaps even simplify the development process [Mooneyhan83].
- The use of a knowledge representation scheme which enables multiple sources of knowledge. A LANDSAT Thematic Mapper image has up to 7 distinct channels. In order to make effective use of spectral values, statistics on segments etc..., we had to find a flexible way of representing these data. The different processes will compute new information on the segments, such as the spatial relations between segments and the spatial classifications of these segments. Finally, LOUSTIC may be interfaced later with other existing systems. How do we add this new knowledge? We chose the frame representation scheme as defined in 1.1.3 as it provides the necessary expandability.

- Segmentation and interpretation are performed at the same stage. Let us recall that an ideal segmentation algorithm does not exist. The result of a segmentation process is characterized by "undersegmented" regions and "oversegmented" regions. "Oversegmented" means the segmentation algorithm was too fine. "Undersegmented" segments result from too coarse a segmentation. However, this process reduces the total amount of data and provides useful hints or areas of interest to the interpretation module.

The idea of the combined segmentation/interpretation process is to refine the initial segmentation with the help of rules defined by an expert. These rules depend on the kind of objects the system must look for. For example, if LOUSTIC tries to search for fields in the image, a segmentation rule will verify if large segments have not been "undersegmented". If so, the refinement consists in splitting this large "undersegmented" segment into two smaller segments.

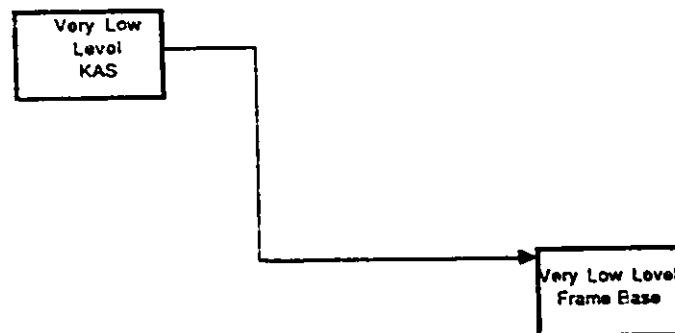
- The use of both bottom-up and top-down control scheme for the interpretation. In LOUSTIC we try to use some of the ideas of Matsuyama on evidence accumulation and top-down search as implemented in SIGMA [Matsuyama86].

- The possibility for a human expert to interact with the system at any time during the execution. Multispectral data are available for the entire Earth on a regular repetitive basis from a number of satellite systems. These data have been archived for ten or more years [Mooneyhan83]. Some static features in the picture may not need to be relocated or interpreted, thus a human expert should have the opportunity to add some further knowledge about the segments. As well the user may want to add new rules to detect other objects present in the scene.

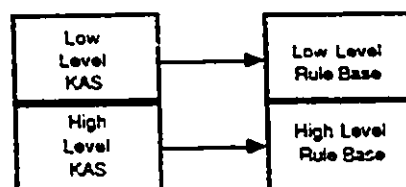
2.2 System Organization

Fig 2.2.1 shows the architecture of LOUSTIC. The three vertical modules on the left constitute the knowledge acquisition system (KAS). The two modules at the centre represent the rule base as entered by the user. The low level rules refer to the symbolic segmentation rules while the high level rules contain the knowledge on the objects the system has to locate. On the right is the frame database where the knowledge on segments is embedded.

•The input LANDSAT picture is first selected by the user. The Very Low Level Knowledge Acquisition System (VLLKAS) refers to the set of low level Fortran routines that segment the image and extract some image features. The VLLKAS first segments the image, generates some statistics on the segments, then each segment is translated into a frame. The set of frames constitutes the Very Low Level Frame Base (VLLFB).



• Then the user inputs the low level rules for segmentation purposes and the high level rules for interpretation. This is done via the two corresponding knowledge acquisition systems.



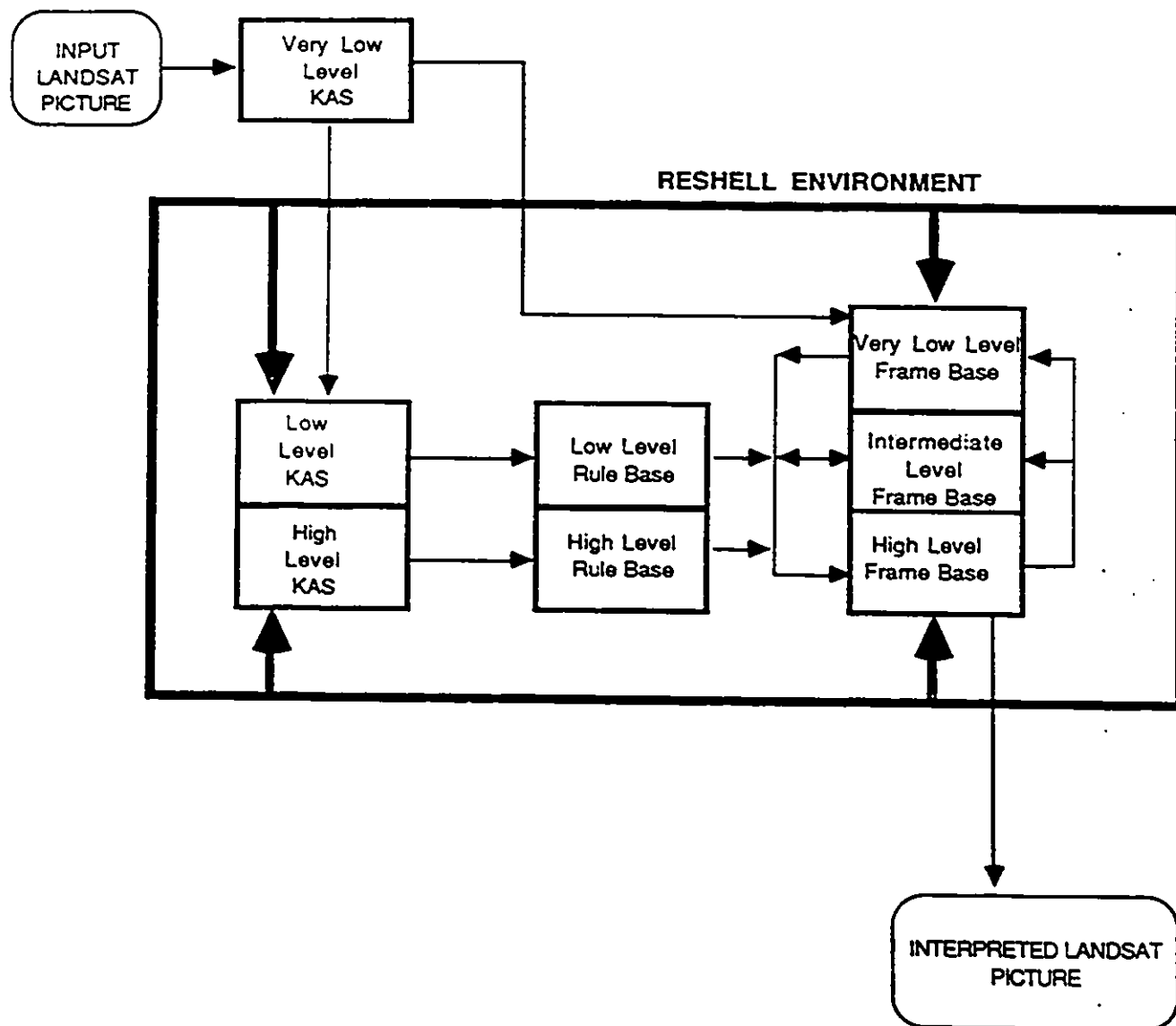
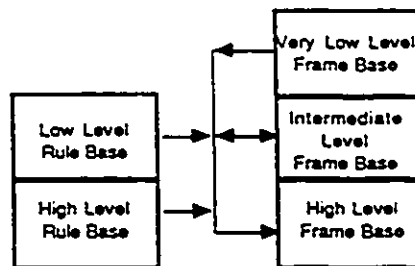
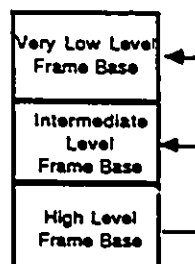


Fig 2.2.1
 SYSTEM ORGANIZATION
 of LOUSTIC

• Applying the low level rules on the very low level frame base leads to the intermediate level frame base (ILFB). At this stage, the system has analyzed the segments, some of which have been split, others of which have been merged. The system is then ready to interpret these segments. The high level frame base refers to all the segments which have been interpreted by LOUSTIC.



• One of the advantages of this frame representation is that a high level frame segment is also an intermediate or low level frame segment . Thus, at any time a frame segment may be reanalyzed or reinterpreted.



• At any time during the process, the user can interact with the frame base or the knowledge acquisition systems via Reshell (Big black arrows in Fig 2.2.1). This allows the user to add knowledge on a particular segment or to consult, display or modify segments or attributes of segments.

In the following subsections, we present each module of LOUSTIC as well as the control strategies and structures in more detail.

2.3 The Low Level Vision System

2.3.1 The Very Low Level Knowledge Acquisition System

The architecture of the VLLKAS is depicted in Fig 2.3.1. Its purpose is to create the VLLFB. This system is similar to the one used in the expert system MICE [Plunkett86] . The image preprocessing begins with a gradient operator (SEGGRA) applied on the input LANDSAT picture. This gradient image is then segmented (SEGMNT). The Sobel gradient operator was chosen because it gave the best results for MICE. The segmentation operator generates a large number of segments from the gradient image and these are then run-length encoded (UNITRU). The image spatial attributes are then converted to a symbolic file (TRUSYM). The segment location file is input to a program which generates a segment spectral statistics file (GSTAT). This file is then converted to symbolic form (STISYM). The resulting symbolic image locations file and symbolic image statistical file are then input to two programs (SYMFRA) and (ADDLINKS). SYMFRA translates each symbolic object segment into a symbolic frame segment while ADDLINKS adds spatial slots for each segment of the frame database.

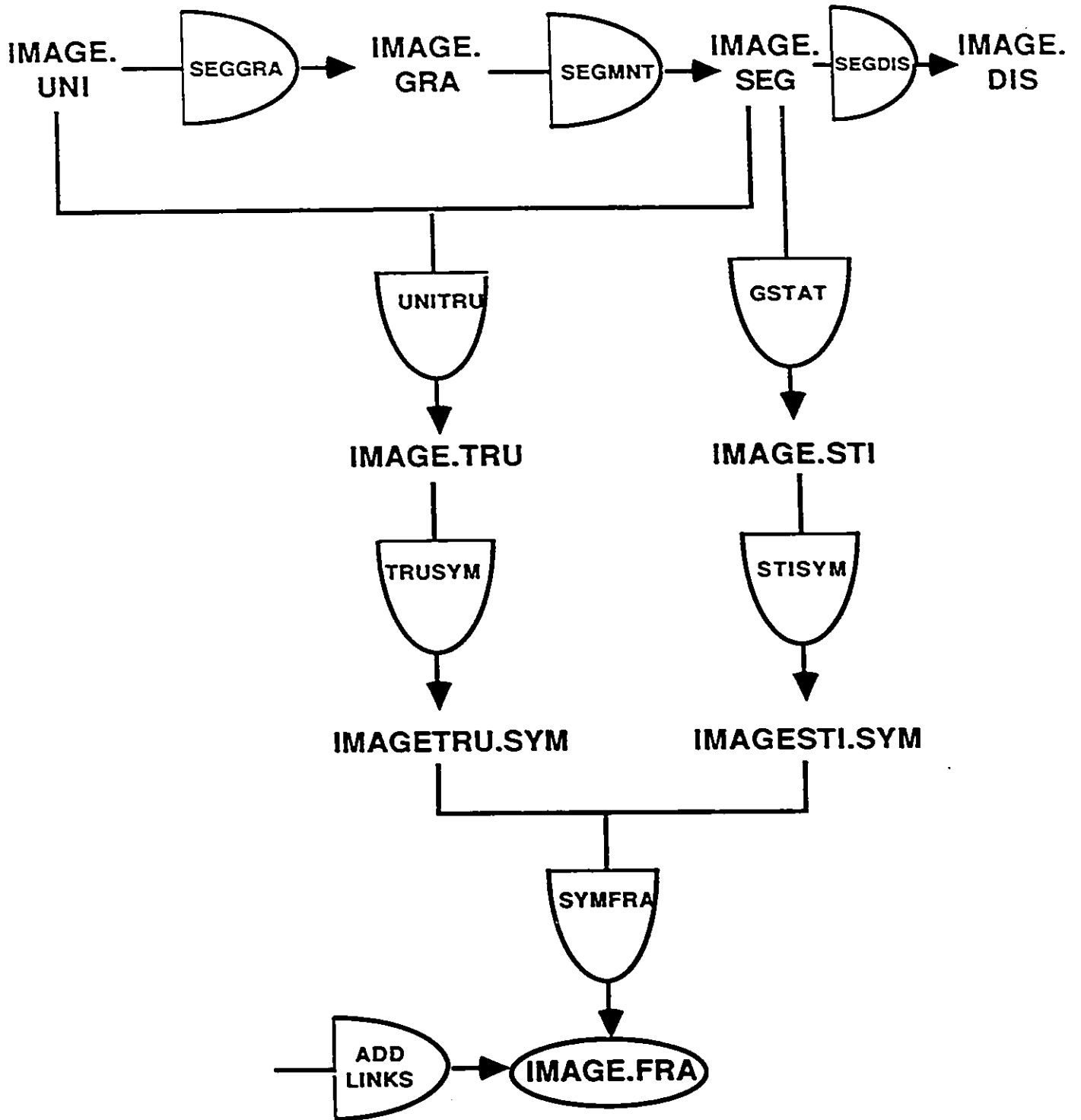


Fig 2.3.1

The format of a frame segment is as follows in semi-Prolog notation.

```

frame : segment_02.
slots : {
  [location,"value",[[1,3,29],[2,3,30],[3,3,33],[4,3,34],[5,3,33],
  [6,4,33],[7,5,33],[8,5,33],[9,1,2],[9,5,33],[10,1,33],[10,40,40],
  [11,1,41],[12,1,42],[13,1,42],[14,1,41],[15,1,41],[16,1,41],
  [17,1,40],[18,1,40],[19,1,40],[20,1,40],[21,1,41],[22,1,42],
  [22,46,47],[23,1,42],[23,45,48],[24,1,41],[24,45,48],[25,1,41],
  [25,45,48],[26,1,42],[26,45,47],[27,1,46],[28,1,46],[29,1,46],
  [30,1,46],[31,1,40],[31,43,45],[32,1,10],[33,1,10],[34,1,10],
  [35,1,10],[36,1,10],[37,1,10],[38,1,10],[39,1,10],[40,1,10],
  [41,1,10],[42,1,10],[43,1,10],[44,1,10],[45,1,10],[46,1,10],[47,1,10]]],
  [shape_1,"value",3.487E1],
  [window,"value",[1,1,47,48]],
  [size,"value",1363],
  [mean_ch_1,"value",8.418E1],
  [mean_ch_2,"value",3.351E1],
  [mean_ch_3,"value",3.642E1],
  [mean_ch_4,"value",3.621E1],
  [mean_ch_5,"value",3.621E1],
  [mean_ch_6,"value",5.479E1],
  [max_ch_1,"value",94],
  [max_ch_2,"value",41],
  [max_ch_3,"value",48],
  [max_ch_4,"value",123],
  [max_ch_5,"value",117],
  [max_ch_6,"value",80],
  [min_ch_1,"value",71],
  [min_ch_2,"value",30],
  [min_ch_3,"value",25],
  [min_ch_4,"value",27],
  [min_ch_5,"value",61],
  [min_ch_6,"value",24],
  [ako,if_needed,compute_ako],
  [segments_linked,if_needed,compute_segments_linked],
  [classify_segments,if_needed,compute_classify_segments]
}.

```

The attributes selected for use by LOUSTIC are the same as the ones selected for MICE:

- Spatial attributes:

1. **LOCATION** : the location attributes represents the location of the pixels in the image segment, based on some reference grid. The value of the location attribute is a three-tuple list that uniquely identifies the location or position of the segment on a line by line basis. The three-tuple contains:

- the line number containing the pixels,
- the start pixel number,
- the end pixel number.

Thus, a list of these run-length-encoded three-tuples will define the location of a segment. A sample segment with its corresponding location is depicted in Fig 2.3.2.

2. **SHAPE**: the shape attribute represents the shape of the segment. The value of the shape is the perimeter squared divided by the area. This shape attribute is a fairly primitive representation of the shape, but additional shape attributes can easily be added if needed. A sample segment with its corresponding shape attribute is depicted in Fig 2.3.2.

3. **WINDOW** : The window attribute represents the smallest rectangle that can be placed around the entire segment. The value of the window attribute is a list of four elements that represent the line and pixel locations of the upper left corner of the window, and the lower right corner. A sample segment with its corresponding window attribute is depicted in Fig 2.3.2.

4. **SIZE** : The size attribute represents the size of the segment. The value of the size attribute is the total number of pixels in the segment. A sample segment with its corresponding size is depicted in Fig 2.3.2.

• Spectral attributes:

The spectral attributes selected for LOUSTIC processing are as follows:

1. **MEAN_CH_1 to MEAN_CH_6**: These attributes represent the mean grey level value of all pixels in the segment for channel 1 to channel 6 of the TM image.

2. **MAX_CH_1 to MAX_CH_6** : These attributes represent the maximum pixel grey level values in the segment for channel 1 to channel 6 of the TM image.

3. **MIN_CH_1 to MIN_CH_6** : These attributes represent the minimum pixel grey level values in the segment for channel 1 to channel 6 of the TM image.

These spectral values are by no means an exhaustive list for classification determination, but they do provide a basis upon which other attributes can be added [Plunkett86].

• Spatial relations between segments:

The purpose of these attributes is to compute when needed the spatial relations between segments.

1. **SEGMENTS_LINKED**: This daemon will search for all the segments whose windows intersect with the currently analyzed frame segment

Fig 2.3.3.

2.CLASSIFY_SEGMENTS: This daemon takes all the segments_linked as computed by the previous daemon and classifies them according to the classifications depicted in Fig 2.3.4.

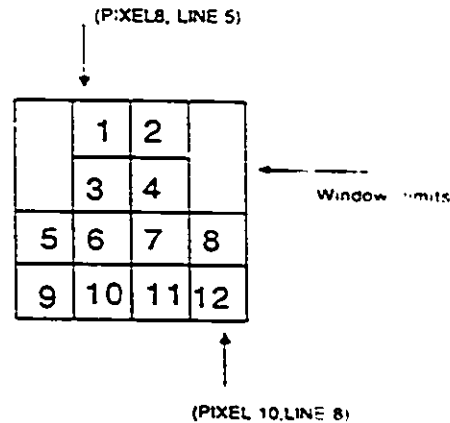


Fig 2.3.2

Spatial Attributes of a segment

LOCATION = [[LINE, START PIXEL, END PIXEL],...]
 = [[5,8,9],[6,8,9],[7,7,10],[8,7,10]]

SHAPE = (PERIMETER ** 2) / AREA
 = ((16 * 50) ** 2) / (50 * 50 * 12)
 = 21.3

SIZE = PIXELCOUNT
 = 12

WINDOW = [UPPER LEFT LINE, UPPER LEFT PIXEL,
 LOWER RIGHT LINE, LOWER RIGHT PIXEL]
 = [5,7,8,10]

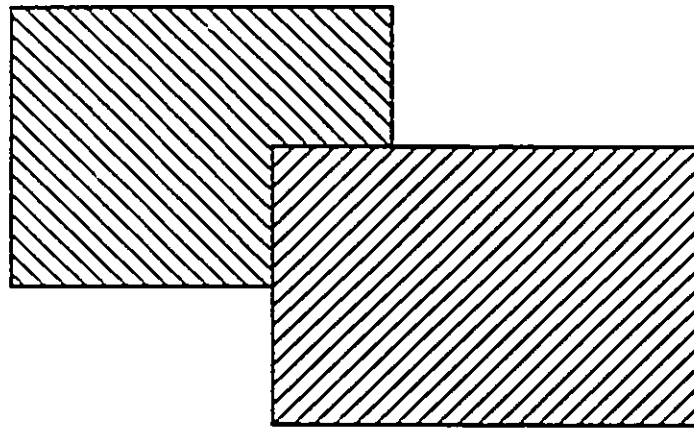


Fig 2.3.3
Segments linked
Intesection of windows

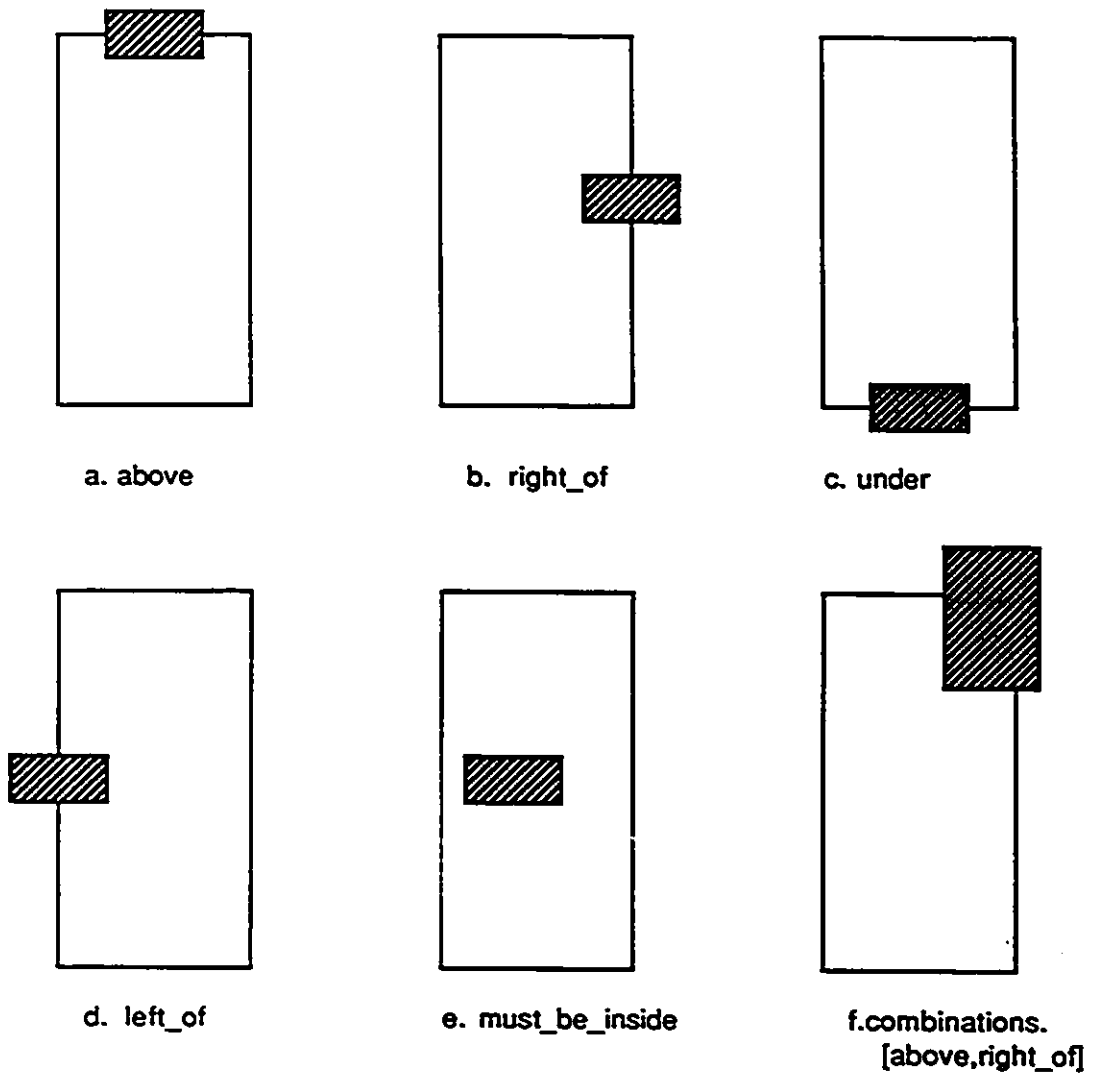


Fig 2.3.4
Spatial relations between segments

2.3.2 The Low Level Knowledge Acquisition System

Following the Very Low Level Knowledge Acquisition System (VLLKAS), the user enters the Low Level Knowledge Acquisition System (LLKAS). A set of merging and splitting rules help refine the original segmentation. These rules are goal directed in a sense because it is one of the objectives of the system to perform Segmentation and Interpretation at the same stage of the process.

There are 4 merging rules:

- IF Size of the small segment is less than or equal to 10

and the means of the small segment are in the minimum to maximum range of the large segment,

THEN merge the two segments.

- IF Size of the small segment is less than or equal to 10

and at least one of the means of the small segment is not in the minimum to maximum range of the large segment,

THEN the small segment cannot be merged with the the large segment.

- IF Size of the small segment is greater than 10

and the sum of the absolute values of the differences of the means on the six channels is less than or equal to 15.

THEN merge the two segments.

- IF Size of the small segment is greater than 10

and the sum of the absolute values of the differences of the means on the six channels is greater than 15.

THEN the small segment cannot be merged to the large segment.

The underlined coefficients may be modified by the user . They characterize the goal directed segmentation/interpretation. Suppose LOUSTIC tries to locate some fields in the LANDSAT picture. A segment which is a potential candidate for a field will have a large number of pixels. The very small segments will not be potential candidates for a field and may result from an oversegmentation. Thus the first and third merging rules try to remove small unneeded segments. If those small segments do not result from an oversegmentation; that is, they represent another feature, rule 2 or 4 is fired and this segment is prevented from merging with a large segment.

There is one splitting rule:

- IF length of the window is greater than 10
and twice the length of the location line does not fit in the window
and the length of the window of the potential new segment is greater than 5 ,
THEN split the segment into two segments.

This splitting rule splits a segment potential candidate for a field into two segments if a certain number of pixels of the analyzed segment does not fill the window. The most promising geometric appearance for a field is a rectangle. If the currently analyzed segment presents some major discontinuities , many unfilled lines, Loustic concludes that there was an "undersegmentation" and splits the segment into two segments.

A detailed example of the execution of these rules is described in section 2.5.

2.4 The High Level Vision System

2.4.1 High Level Knowledge Acquisition System and Bottom-Up Interpretation

As the bottom-up interpretation scheme is data driven, the system needs some knowledge about the objects to perform its task. In LOUSTIC, prior to the segments analysis, the user enters knowledge on the objects he wants the system to locate. An example of a bottom-up interpretation rule for a field as defined by a user is :

- IF Length of the segment is at least 10
and Width of the segment is at least 10
and Size of the segment is at least 100
and the majority of the lines fit in the window,
THEN the segment is interpreted as a field in a bottom-up way.

The underlined coefficients are the ones entered by the user. The LANDSAT Thematic Mapper image has a pixel size of 25 meters when geometrically corrected. A segment whose attributes match the conditions of this rule will be interpreted as a field of minimum size 250 meters by 250 meters. This corresponds to small fields in Saskatchewan.

2.4.2 Bottom-up Interpretation using the Evidence Accumulation Approach:

LOUSTIC also proposes a bottom-up interpretation scheme if the previous bottom-up approach is without results. This approach is similar to the one developed by Matsuyama in SIGMA [Matsuyama86] although not implemented in the same manner. The appearance of objects is represented in SIGMA with a network where the objects are

described by nodes and the relations by arcs. A model of a suburban housing development was presented in Fig 1.3.3.3. To build a similar network for a satellite picture of Saskatchewan, we can use the following assertions:

- A field is normally close to other fields.
- Roads are eventually at the intersection of fields.
- Houses are close to roads.
- The appearance of a field is a large rectangle.
- The appearance of a parts of road is a small elongated rectangle.
- Houses are little spots in satellite pictures.

The resulting network is depicted in Fig 2.4.2.1

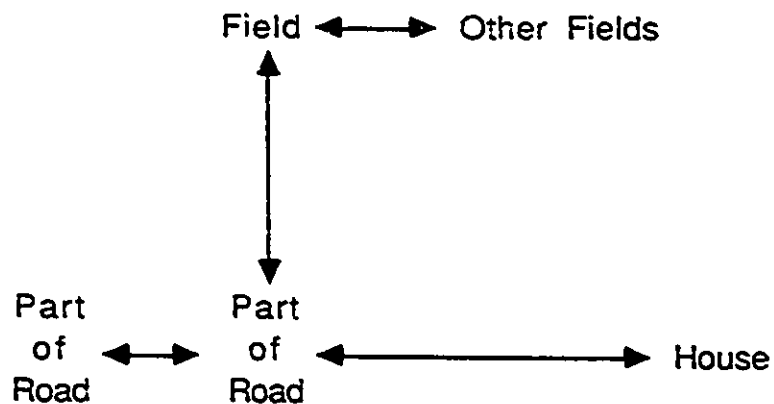


Fig 2.4.2.1

Spatial links of objects
in a satellite picture

On accumulation of evidence, LOUSTIC does not use rules to confirm hypotheses. Instead, it looks in the neighborhood of a particular segment to check for the interpretation of other segments and decides whether or not a segment is an instance of an object. For example, suppose the bottom-up approach described in section 2.4.1 fails to interpret a segment in a pure bottom-up way; LOUSTIC starts the evidence accumulation approach and searches for the surrounding segments and their interpretation. If a certain number of these surrounding segments has been instantiated as fields, then the currently analyzed segment is interpreted as a field. For example, if a large segment is surrounded by at least two segments interpreted as fields, LOUSTIC concludes that the current segment is a field.

2.4.3 Top-Down Interpretation

The top-down interpretation scheme is fired when both bottom-up approaches fails. In this case LOUSTIC sends the user back to the LLKAS and the user has the opportunity to change the coefficients of the merging and splitting rules. If he does, these new rules are made local to the currently analyzed segment and a new segmentation/interpretation process can be started. Once this process is done, the system retrieves the previous segmentation rules.

The entire high level process is depicted in Fig 2.4.1.

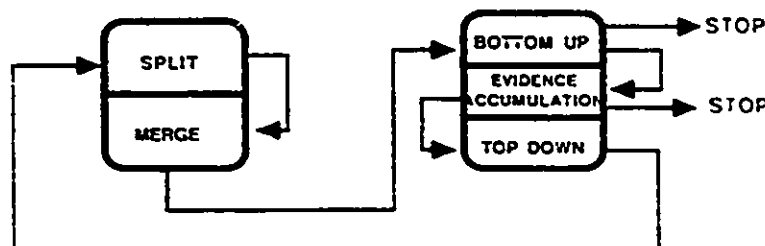


Fig 2 4 1
Interpretation stage: the strategies

2.5 Scheduling and control, the construction of interpretation

The schematic diagram of the processing involved in constructing interpretations is shown in Fig 2.5.1. The following steps are iterated in the process:

- focus of attention,
- analysis,
- interpretation.

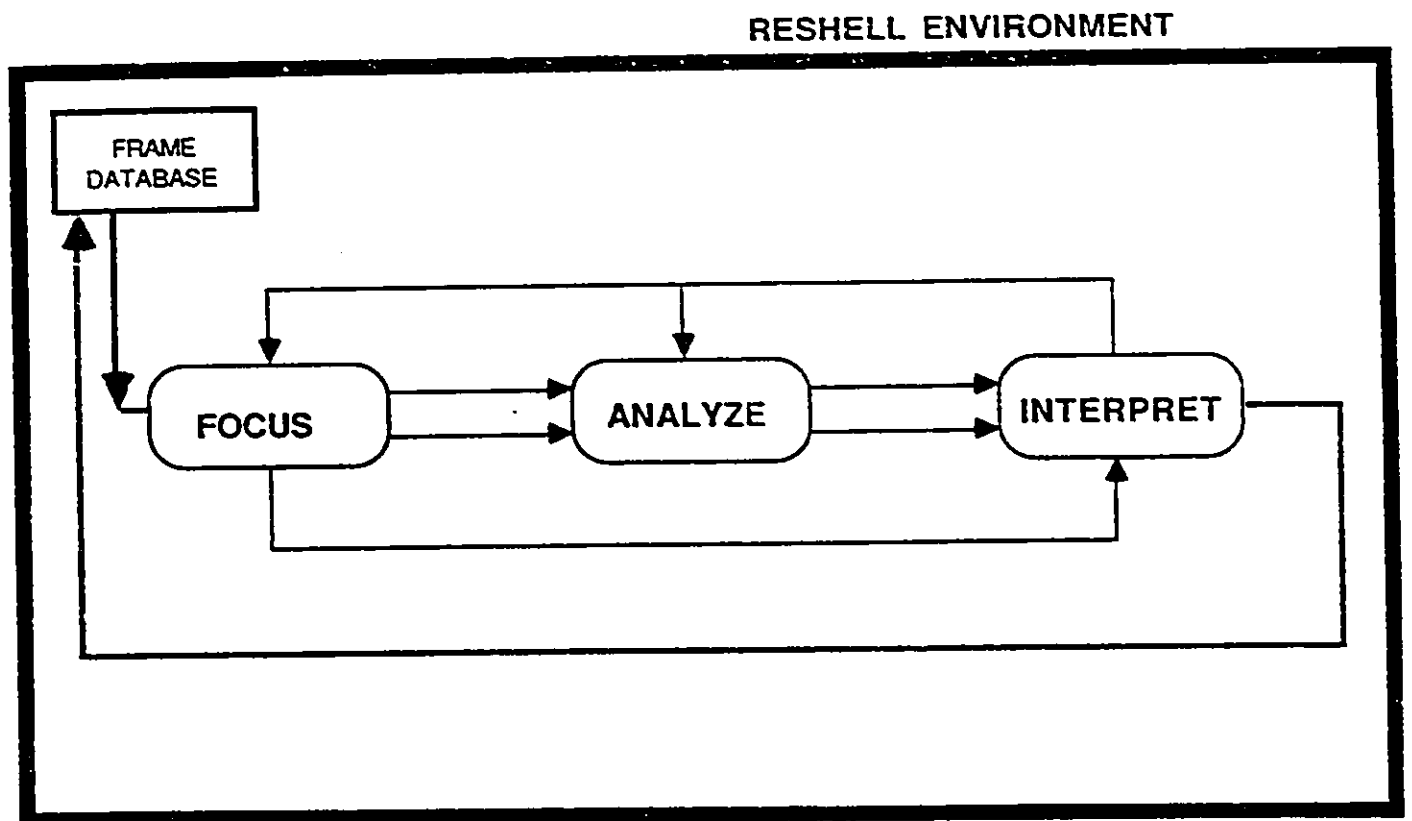
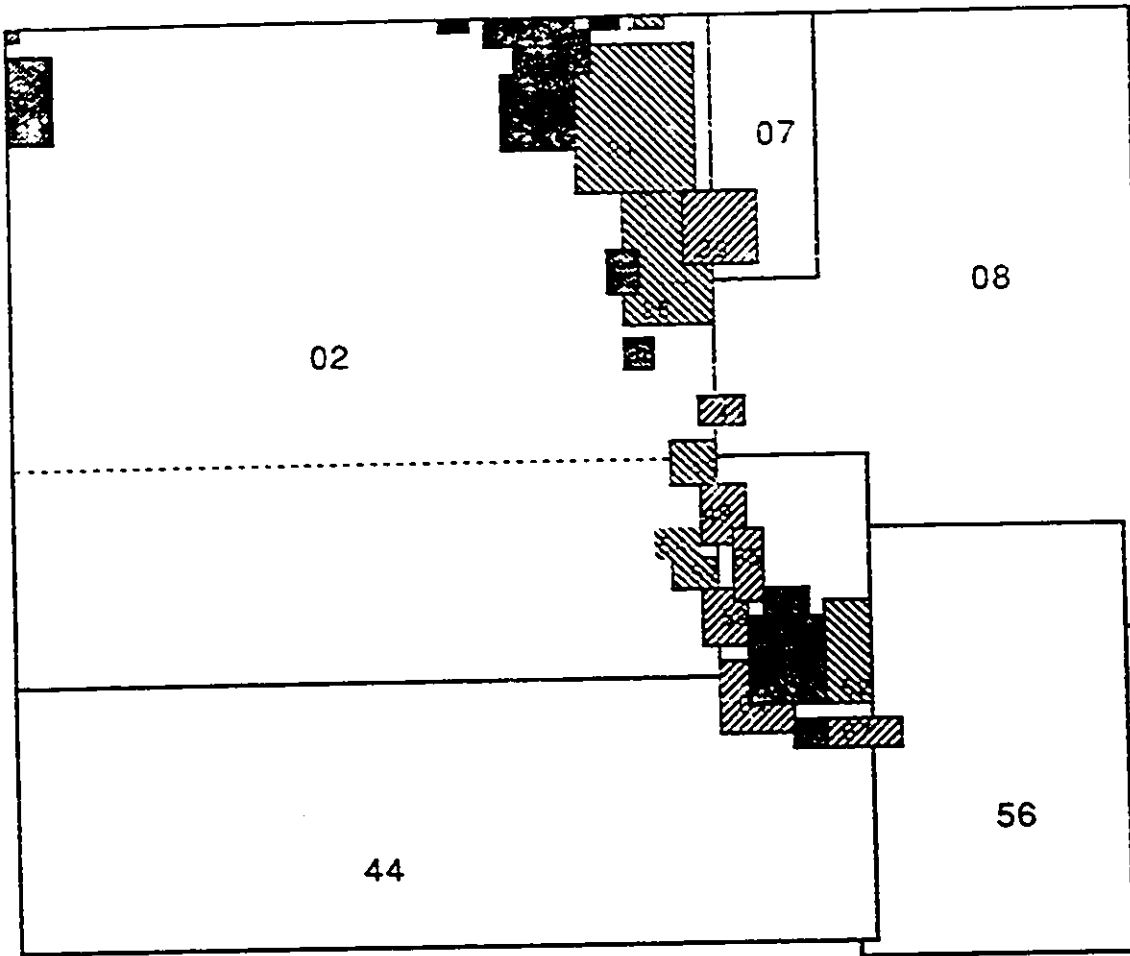


Fig 2.5.1
SCHEDULER

- **The focus of attention mechanism** gives the user two different alternatives as seen in Fig 2.5.1.
 - Focus the attention of LOUSTIC on the frame segments which are potential candidates for fields as selected by LOUSTIC according to the knowledge it has on fields.
 - Focus the attention of the system on every segment of the frame database.

- **Analysis** . Once a segment is selected. LOUSTIC analyses this particular segment. This analysis is performed in 5 different steps:
 - Apply the splitting rule to the current segment.
 - Search all segments whose window intersects with the window of current segment.
 - Apply the merging rules to all these intersecting segments.
 - Merge all segments which are inside the current segment and satisfy the conditions of the merging rules.
 - Prompt the user for further merging : does he wants to merge other possible mergeable segments to the current segment .

Fig 2.5.2 presents the window of some segments as selected in the prototype picture. Segments 02, 07, 08, 44 and 56 are potential candidates for fields according to LOUSTIC. The other segments are all the segments whose windows intersect with the windows of segment_02 and segment_44.



Small segment inside a large segment



Small segment belonging to two large segments



Small segment inside a large segment and intersecting another large segment



Dotted lines represent the common part of two segments

Fig 2.5.2
Segment_02, Segment_04
and intersecting segments

Assume the user asks LOUSTIC to focus the attention on segment_02. The frame segment_02 before the analysis is as follows:

```

frame : segment_02.
slots : [
  [location,"value",[[1,3,29],[2,3,30],[3,3,33],[4,3,34],[5,3,33],
  [6,4,33],[7,5,33],[8,5,33],[9,1,2],[9,5,33],[10,1,33],[10,40,40],
  [11,1,41],[12,1,42],[13,1,42],[14,1,41],[15,1,41],[16,1,41],
  [17,1,40],[18,1,40],[19,1,40],[20,1,40],[21,1,41],[22,1,42],
  [22,46,47],[23,1,42],[23,45,48],[24,1,41],[24,45,48],[25,1,41],
  [25,45,48],[26,1,42],[26,45,47],[27,1,46],[28,1,46],[29,1,46],
  [30,1,46],[31,1,40],[31,43,45],[32,1,10],[33,1,10],[34,1,10],
  [35,1,10],[36,1,10],[37,1,10],[38,1,10],[39,1,10],[40,1,10],
  [41,1,10],[42,1,10],[43,1,10],[44,1,10],[45,1,10],[46,1,10],[47,1,10]]
  [shape_1,"value",3.487E1],
  [window,"value",[1,1,47,48]],
  [size,"value",1363],
  [mean_ch_1,"value",8.418E1],
  [mean_ch_2,"value",3.351E1],
  [mean_ch_3,"value",3.642E1],
  [mean_ch_4,"value",3.621E1],
  [mean_ch_5,"value",3.621E1],
  [mean_ch_6,"value",5.479E1],
  [max_ch_1,"value",94],
  [max_ch_2,"value",41],
  [max_ch_3,"value",48],
  [max_ch_4,"value",123],
  [max_ch_5,"value",117],
  [max_ch_6,"value",80],
  [min_ch_1,"value",71],
  [min_ch_2,"value",30],
  [min_ch_3,"value",25],
  [min_ch_4,"value",27],
  [min_ch_5,"value",61],
  [min_ch_6,"value",24],
  [ako,if_needed,compute_ako],
  [segments_linked,if_needed,compute_segments_linked],
  [classify_segments,if_needed,compute_classify_segments]
].

```

- Splitting step:

The window of segment_02 is [1,1,47,48] and segment_02 satisfies the conditions of the splitting rule Fig 2.5.3. Thus segment_02 is split into two segments as depicted in Fig 2.5.4. A new segment is added to the segments frame data base with its own slots and the analysis of segment_02 is pursued with the new window of segment_02 : [1,1,31,48].

In this example the new segment is segment_71 and its frame is :

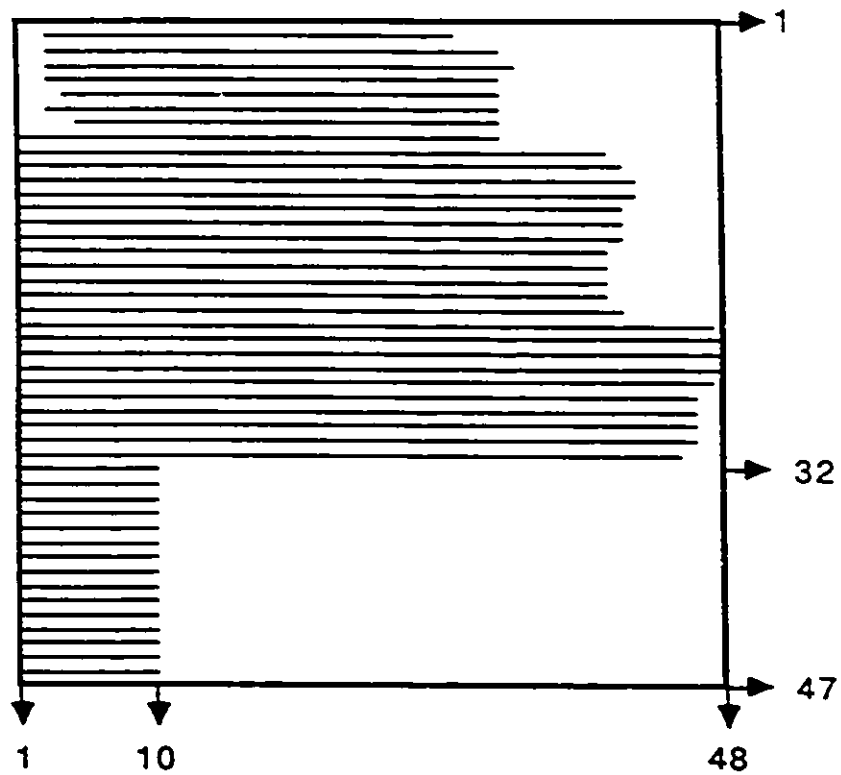
```

frame : segment_71.
slots : [
[location,"value",[[32,1,10],[33,1,10],[34,1,10],[35,1,10],[36,1,10],
[37,1,10],[38,1,10],[39,1,10],[40,1,10],[41,1,10],[42,1,10],[43,1,10],
[44,1,10],[45,1,10],[46,1,10],[47,1,10]]],
>window,"value",[32,1,47,10]],
>min_ch_1,"value",71],
>max_ch_1,"value",94],
>mean_ch_1,"value",8.418E1],
>min_ch_2,"value",30],
>max_ch_2,"value",41],
>mean_ch_2,"value",3.351E1],
>min_ch_3,"value",25],
>max_ch_3,"value",48],
>mean_ch_3,"value",3.642E1],
>min_ch_4,"value",27],
>max_ch_4,"value",123],
>mean_ch_4,"value",3.621E1],
>min_ch_5,"value",61],
>max_ch_5,"value",117],
>mean_ch_5,"value",3.621E1],
>min_ch_6,"value",24],
>max_ch_6,"value",80],
>mean_ch_6,"value",5.479E1],
>segments_linked,if_needed,compute_segments_linked],
>classify_segments,if_needed,compute_classify_segments],
>size,"value",160]
].

```

Fig 2.5.5

New created segment

segment_02**IF**

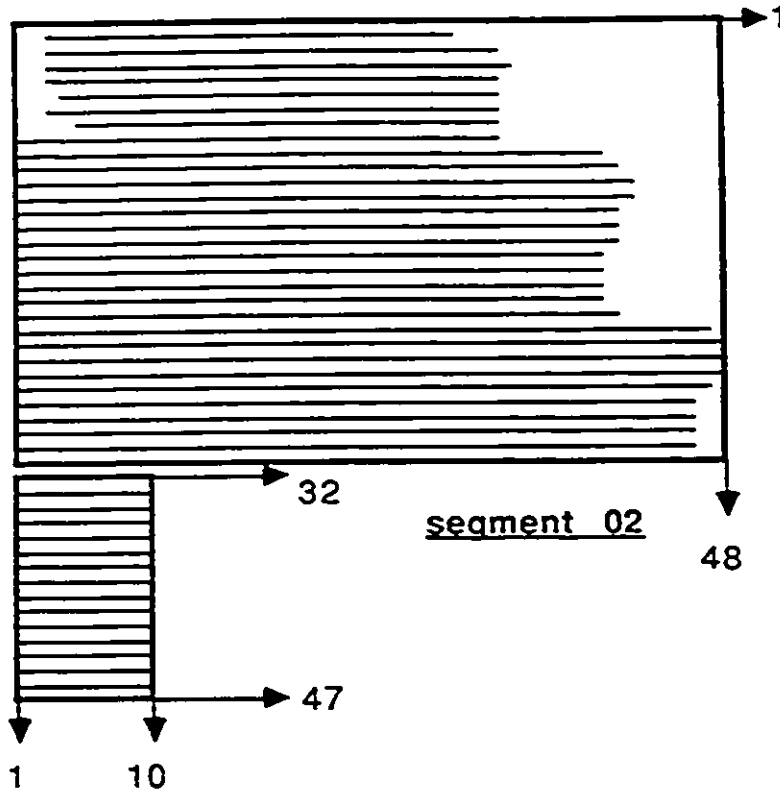
length of the window is greater than 10
 and twice the length of the line
 does not fit in the window
 and if length of the new window
 is greater than 5

(Length = 47)

*(2*10 < 48 at line 10)*

(Length = 16)

Fig 2.5.3
 Example of a split:
 Conditions



New segment

THEN

Split the segment into two parts.

Fig 2.5.4
Example of a split :
Action

- Merging step:

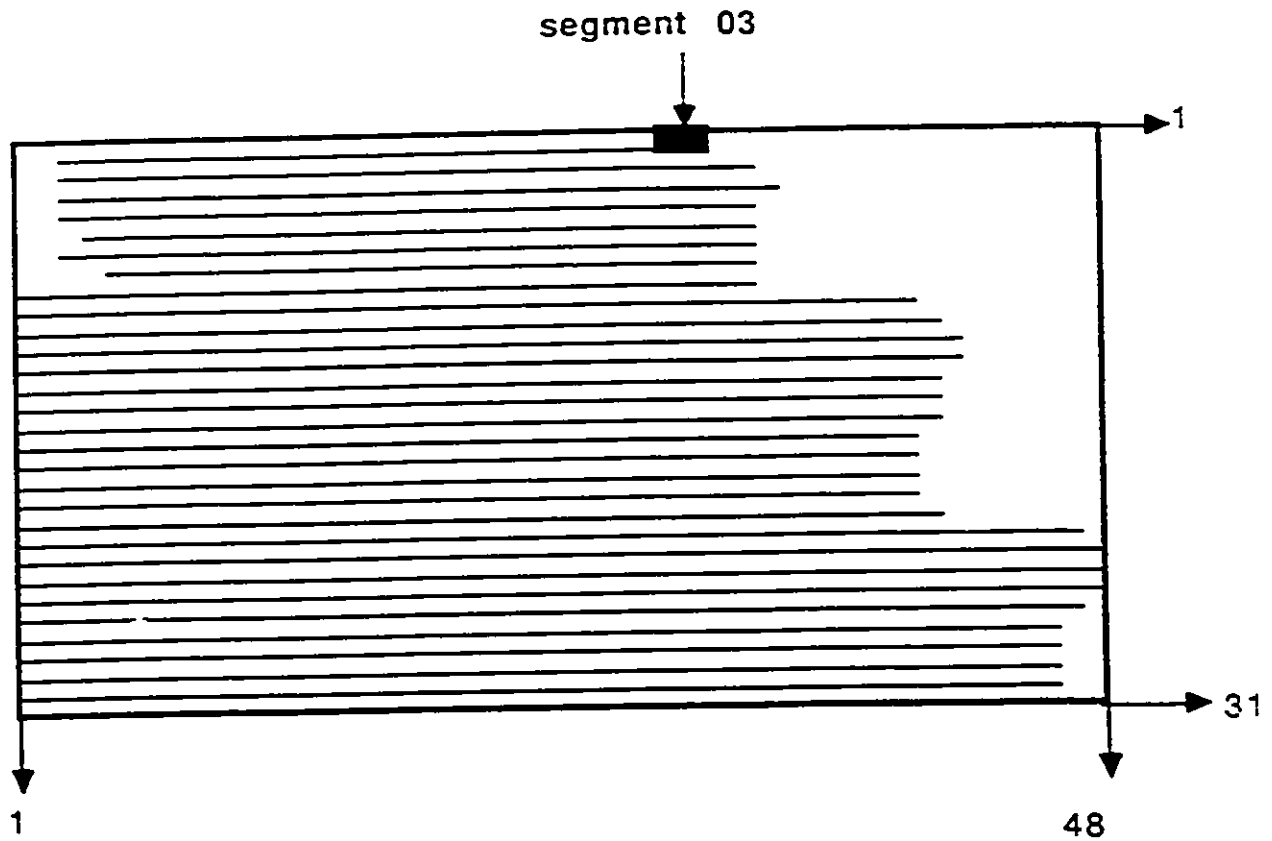
As seen in Fig 2.5.6, segment_03 is a small segment inside segment_02 and it satisfies the conditions of the merging rule as defined by the user in the LLKAS. Therefore segment_03 is merged to segment_02 (Fig 2.5.7) and its frame is updated

```

frame : segment_03.
slots : [
  [location,"value",[[1,30,32],[2,31,32]]],
  [shape_1,"value",20],
  [window,"value",[1,30,2,32]],
  [size,"value",5],
  [mean_ch_1,"value",7.64E1],
  [mean_ch_2,"value",34],
  [mean_ch_3,"value",29],
  [mean_ch_4,"value",9.34E1],
  [mean_ch_5,"value",9.34E1],
  [mean_ch_6,"value",3.44E1],
  [max_ch_1,"value",79],
  [max_ch_2,"value",36],
  [max_ch_3,"value",32],
  [max_ch_4,"value",109],
  [max_ch_5,"value",85],
  [max_ch_6,"value",46],
  [min_ch_1,"value",75],
  [min_ch_2,"value",31],
  [min_ch_3,"value",26],
  [min_ch_4,"value",59],
  [min_ch_5,"value",59],
  [min_ch_6,"value",20],
  [ako,if_needed,compute_ako],
  [segments_linked,if_needed,compute_segments_linked],
  [classify_segments,if_needed,compute_classify_segments]
].

frame : segment_03.
slots : [
  [shape_1,"value",20],
  [mean_ch_1,"value",7.64E1],
  [mean_ch_2,"value",34],
  [mean_ch_3,"value",29],
  [mean_ch_4,"value",9.34E1],
  [mean_ch_5,"value",9.34E1],
  [mean_ch_6,"value",3.44E1],
  [max_ch_1,"value",79],
  [max_ch_2,"value",36],
  [max_ch_3,"value",32],
  [max_ch_4,"value",109],
  [max_ch_5,"value",85],
  [max_ch_6,"value",46],
  [min_ch_1,"value",75],
  [min_ch_2,"value",31],
  [min_ch_3,"value",26],
  [min_ch_4,"value",59],
  [min_ch_5,"value",59],
  [min_ch_6,"value",20],
  [ako,if_needed,compute_ako],
  [segments_linked,if_needed,compute_segments_linked],
  [classify_segments,if_needed,compute_classify_segments],
  [merged_to,"value",[segment_02,'at_step',3]],
  [location,"value",[[0,0,0,0]]],
  [window,"value",[0,0,0,0]],
  [size,"value",0]
].

```



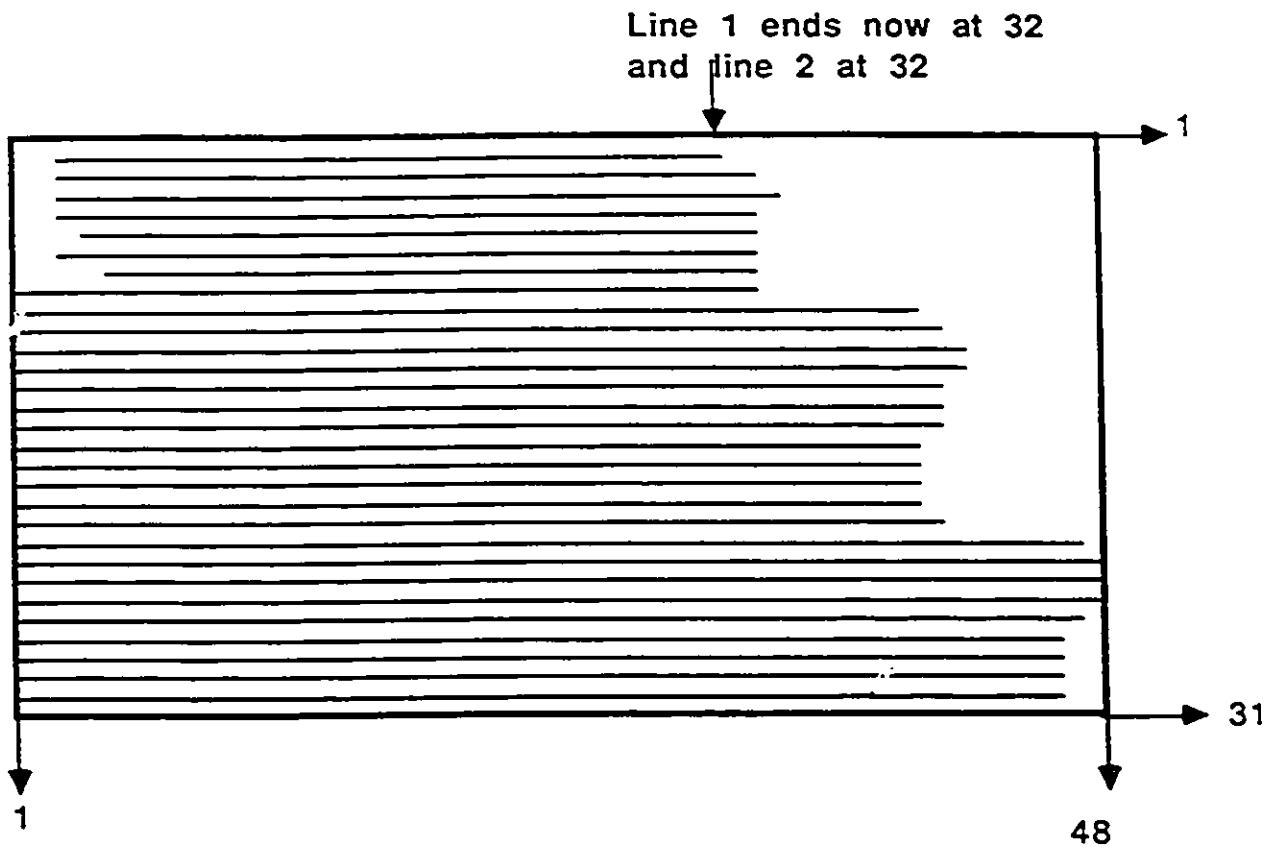
New segment 02

IF

Size of small segment is less than or equal to 10 (Size = 5)
 and the means of the small segment are
 in the min-max range of the large segment

| <i>min</i> | <i>means</i> | <i>max</i> |
|------------|--------------|------------|
| 71 | < 76.4 | < 94 |
| 30 | < 34 | < 41 |
| 25 | < 34 | < 48 |
| 27 | < 93.4 | < 123 |
| 61 | < 93.4 | < 117 |
| 20 | < 34.4 | < 80 |

Fig 2.5.6
 Merge of a small segment :
 Conditions



New segment 02
Segment 03 merged

THEN

Merge the small segment to the large segment.

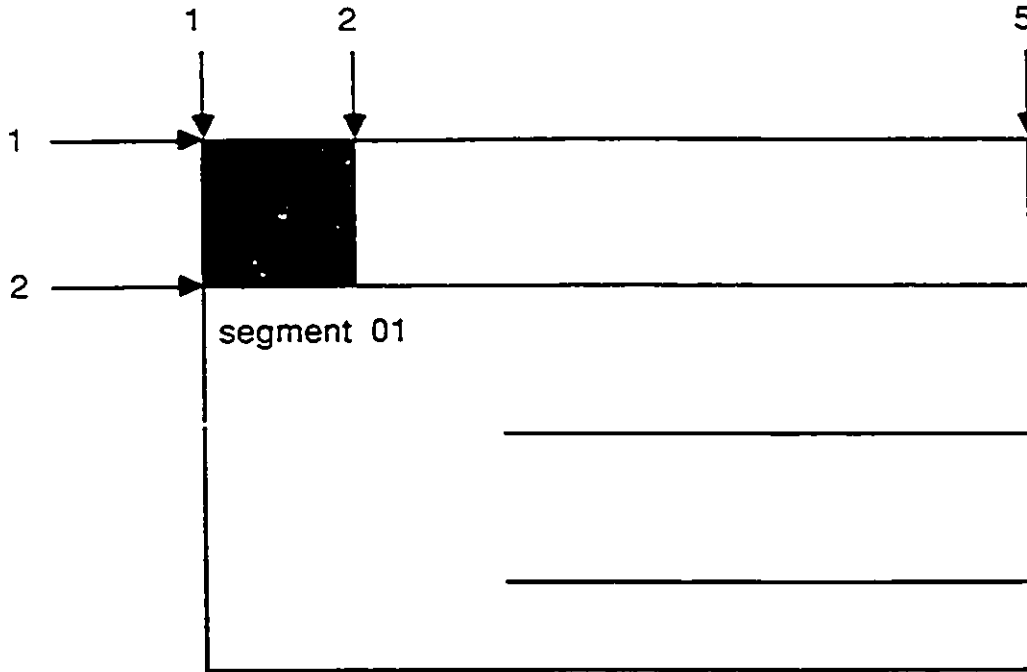
Fig 2.5.7
Merge of a small segment :
Action

Segment_01 is also a small segment inside segment_02. However, its mean on channel 5 is not in the minimum to maximum range of segment_02 (Fig 2.5.8). Thus, segment_01 is not the same object as segment_02 and is prevented from merging with segment_02 (Fig 2.5.9). Segment_01 has now reached the stage of the intermediate level frame representation. That is, it has been analyzed and is ready for interpretation. Its intermediate frame is given below:

```

frame : segment_01.
slots : [
  [location,"value",[[1,1,2],[2,1,2]]],
  [shape_1,"value",16],
  [window,"value",[1,1,2,2]],
  [size,"value",4],
  [mean_ch_1,"value",8.625E1],
  [mean_ch_2,"value",3.55E1],
  [mean_ch_3,"value",3.95E1],
  [mean_ch_4,"value",3.925E1],
  [mean_ch_5,"value",3.925E1],
  [mean_ch_6,"value",6.65E1],
  [max_ch_1,"value",87],
  [max_ch_2,"value",37],
  [max_ch_3,"value",42],
  [max_ch_4,"value",41],
  [max_ch_5,"value",103],
  [max_ch_6,"value",70],
  [min_ch_1,"value",85],
  [min_ch_2,"value",35],
  [min_ch_3,"value",38],
  [min_ch_4,"value",36],
  [min_ch_5,"value",95],
  [min_ch_6,"value",64],
  [ako,if_needed,compute_ako],
  [segments_linked,if_needed,compute_segments_linked],
  [classify_segments,if_needed,compute_classify_segments],
  [not_mergeable_to,"value",[[segment_02,'because_of',min_max],
  [segment_02,'because_of',means]]]
].

```



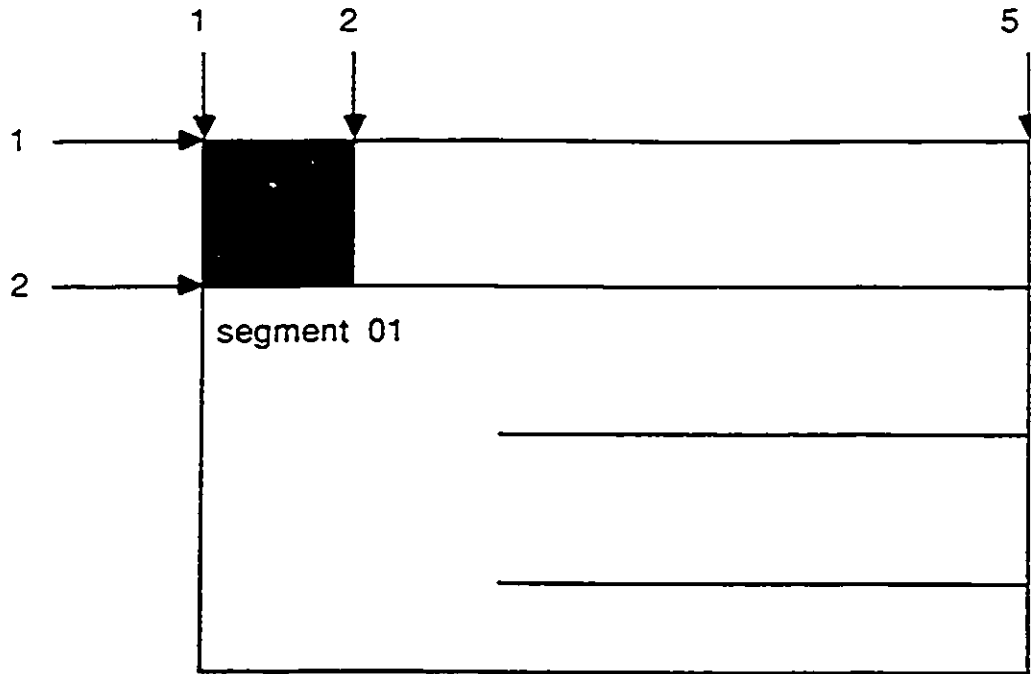
Segment 02
(Lines 1 to 4)

IF

Size of small segment is less than or equal to 10 (Size = 4)
and the means of the small segment are not
in the min-max range of the big segment

| | <i>min</i> | <i>means</i> | <i>max</i> |
|---|------------|--------------|------------|
| | 71 | < 86.25 | < 94 |
| | 30 | < 35.5 | < 41 |
| | 25 | < 39.5 | < 58 |
| | 27 | < 39.25 | < 123 |
| → | 61 | > 39.25 | < 117 |
| | 20 | < 66.5 | < 80 |

Fig 2.5.8
Impossible merge of a small segment : Conditions



Segment 02
(Lines 1 to 4)

THEN

Prevent the small segment from merging with the large segment.

Fig 2.5.9
Impossible merge of a small segment : Action

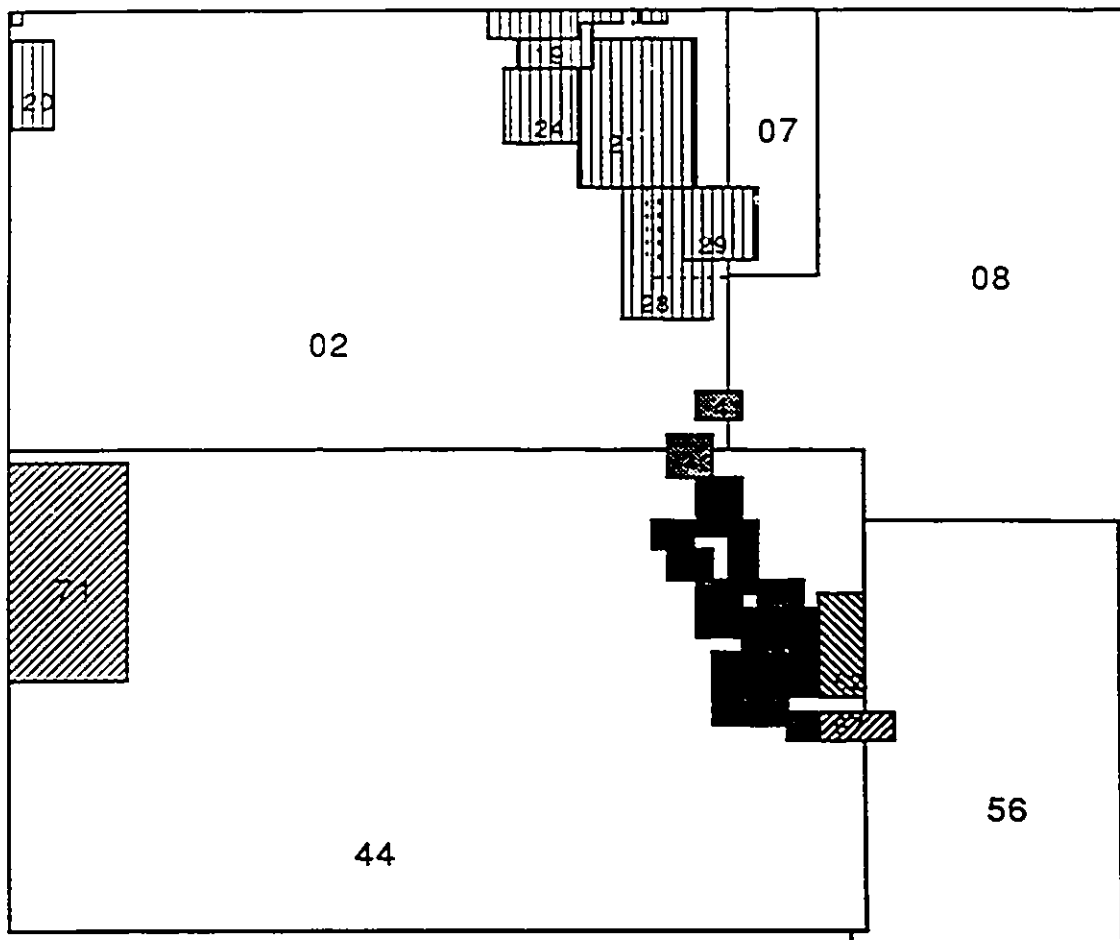
The analysis continues until all the intersecting segments are analyzed. Segment_02 and its surrounding segments are depicted in Fig 2.5.10. As explained before, segment_41 and segment_43 satisfy the conditions for the merging rules. However, since they are not inside segment_02, LOUSTIC then asks the user if he wants to merge these two segments.

Segment_02 after this analysis is given below:

```

frame : segment_02.
slots : [
  [shape_1,"value",3.487E1],
  [mean_ch_1,"value",8.418E1],
  [mean_ch_2,"value",3.351E1],
  [mean_ch_3,"value",3.642E1],
  [mean_ch_4,"value",3.621E1],
  [mean_ch_5,"value",3.621E1],
  [mean_ch_6,"value",5.479E1],
  [max_ch_1,"value",94],
  [max_ch_2,"value",41],
  [max_ch_3,"value",48],
  [max_ch_4,"value",123],
  [max_ch_5,"value",117],
  [max_ch_6,"value",80],
  [min_ch_1,"value",71],
  [min_ch_2,"value",30],
  [min_ch_3,"value",25],
  [min_ch_4,"value",27],
  [min_ch_5,"value",61],
  [min_ch_6,"value",24],
  [ako,if_needed,compute_ako],
  [segments_linked,if_needed,compute_segments_linked],
  [classify_segments,if_needed,compute_classify_segments],
  [size,"value",1203],
  [window,"value",[1,1,31,48]],
  [location,"value",[[1,3,32],[2,3,32],[3,3,33],[4,3,34],[5,3,33],
  [6,4,33],[7,5,33],[8,5,33],[9,1,2],[9,5,33],[10,1,33],[10,40,40],
  [11,1,41],[12,1,42],[13,1,42],[14,1,41],[15,1,41],[16,1,41],[17,1,42],
  [18,1,43],[19,1,43],[20,1,42],[21,1,41],[22,1,42],[22,46,47],[23,1,48],
  [24,1,48],[25,1,48],[26,1,47],[27,1,46],[28,1,46],[29,1,46],[30,1,46],
  [31,1,40],[31,43,45]]],
  [merged_with,"value",[[segment_38,'at_step',1],[segment_32,'at_step',1],
  [segment_03,'at_step',3]]],
  [not_mergeable_to,"value",[[segment_44,'because_of',means],[segment_29,
  'because_of',means],[segment_28,'because_of',means],
  [segment_24,'because_of',means],[segment_22,'because_of',min_max],
  [segment_21,'because_of',means],[segment_20,'because_of',means],
  [segment_19,'because_of',means],[segment_07,'because_of',means],
  [segment_06,'because_of',min_max],[segment_05,'because_of',
  min_max],[segment_04,'because_of',means],
  [segment_01,'because_of',min_max],[segment_01,'because_of',means]]]
].

```









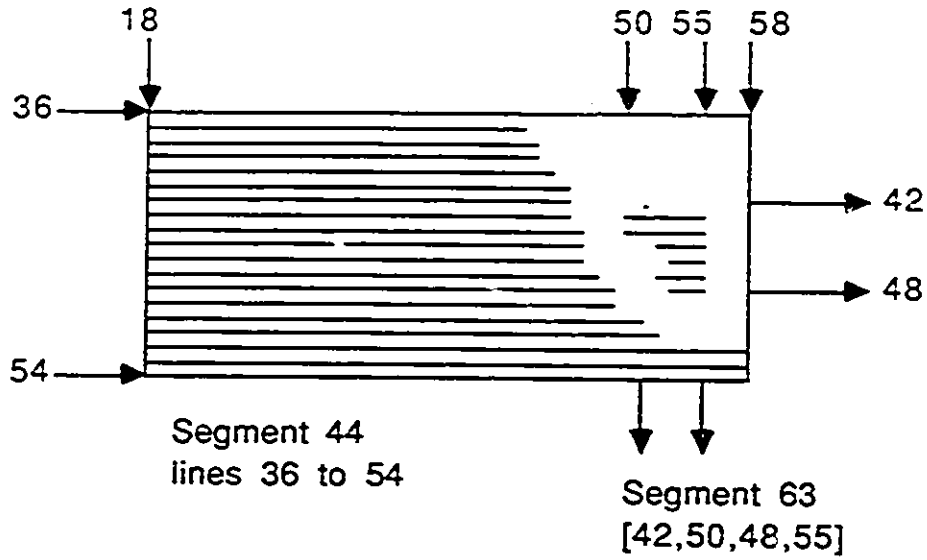
-  small segment inside a large segment
-  small segment inside a large segment but intersecting another large segment
-  small segment belonging to two large segments
-  impossible merge
-  later possible merge
-  new segment

Fig 2.5.10

End of the Analysis for Segment_02

Because segment_02 was split into two parts the status of some segments which intersected with segment_02 before this step changed. These changes can also be observed in Fig 2.5.10. Suppose the user now wants to focus the attention of LOUSTIC on segment_44, Fig 2.5.11 and Fig 2.5.12 present the merge of segment_63 to segment_44. Segment_63 satisfies the conditions of the merging rule for bigger segments (Size > 10). Fig 2.5.13 shows the result of the analysis performed on segment_02 and segment_44. All the unneeded segments are removed, segment_41 and segment_43 may be merged later to either segment_02, segment_08 or segment_44 , and the remaining segments have all reached the stage of the intermediate level of the representation as they cannot be merged to the analyzed segments. Segment_02 and segment_44 have also reached this stage and are ready to be interpreted by LOUSTIC.



IF

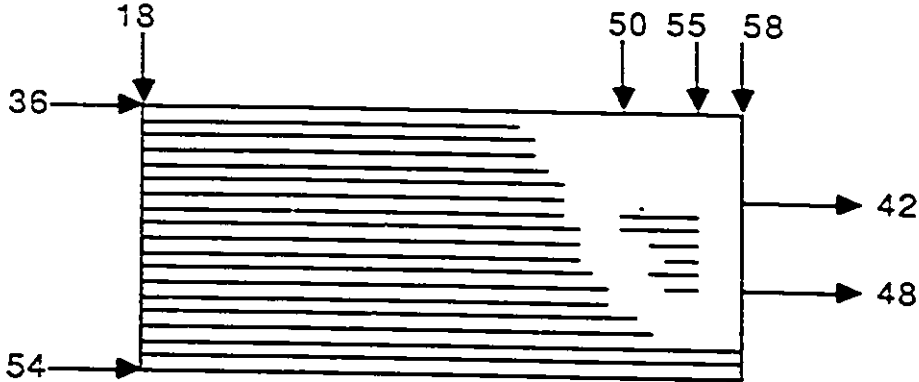
Size of small segment is greater to 10
 and the sum of the difference of the means
 on the 6 channels is less than or equal to 15

(Size = 27)

| <i>mean small</i> | <i>mean big</i> | |
|-----------------------|---------------------|------|
| 77.7 | 76.24 | 1.46 |
| 31.9 | 31.33 | 0.57 |
| 27.8 | 27.01 | 0.79 |
| 85 | 87.69 | 2.69 |
| 85 | 87.69 | 2.69 |
| 21.07 | 19.98 | 1.69 |
| | | 9.89 |

+

Fig 2.5.11
 Merge of a segment:
 Conditions

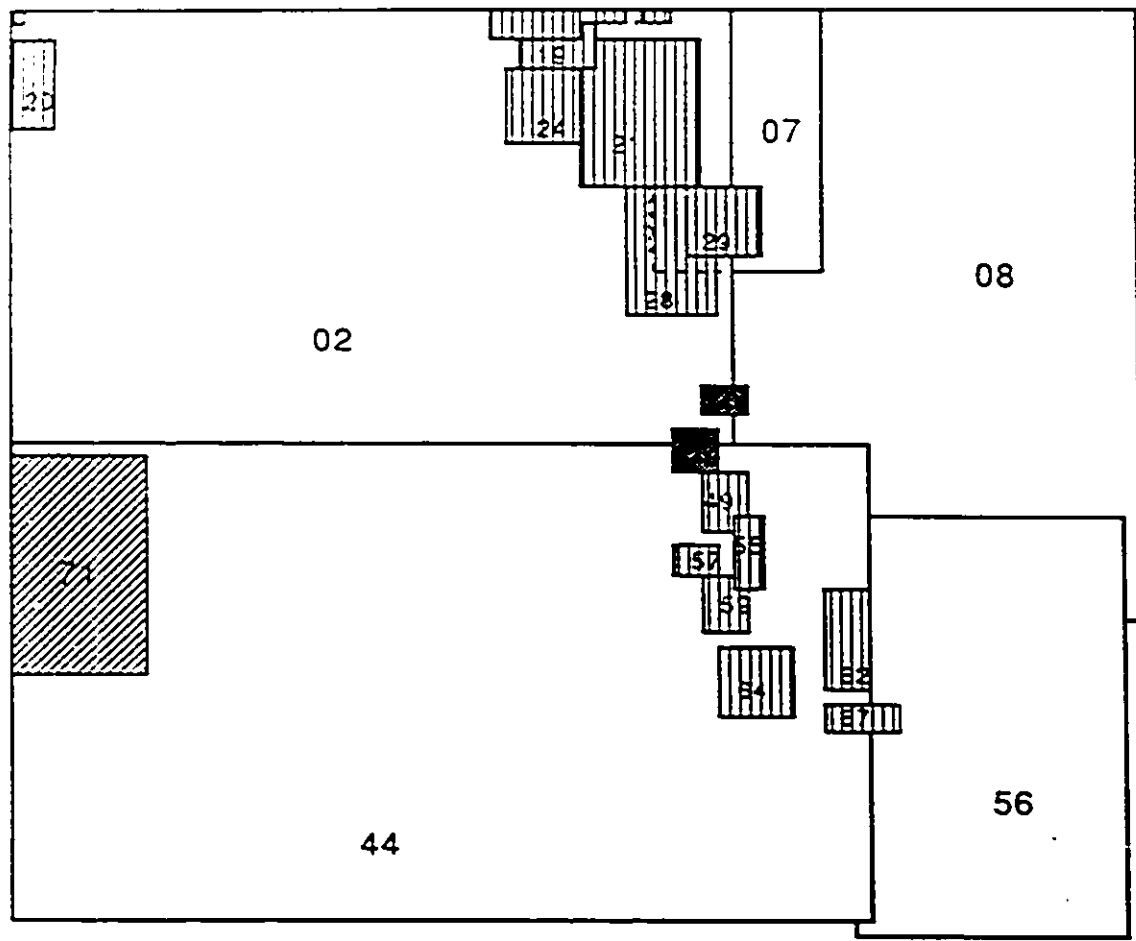


Segment 44
lines 36 to 54
Segment 63
has been merged

THEN

Merge the two segments.

Fig 2.5.12
Merge of a segment:
Action






-  impossible merge
-  later possible merge
-  new segment

Fig 2.5.13
End of the analysis for
Segment_02 and Segment_44

- Interpretation:

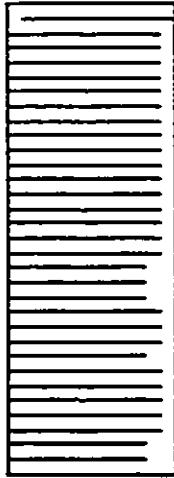
Once a segment has reached the stage of the intermediate level of representation, it may be interpreted by LOUSTIC. LOUSTIC always starts its interpretation using the bottom-up approach. Fig 2.5.14 depicts segment_94, a segment which satisfies the conditions of the bottom-up rules for being a field: its width is greater than 10, its length is greater than 10, its size is greater than 100, and for its main part, its lines fill the window with a margin of error of 2 pixels for each line. Therefore, it is interpreted as a field in a bottom-up way

(Fig 2.5.15). Two slots are added into its frame :

- interpreted_as which gives the interpretation.
- interpretation gives the approach used for this interpretation.

The frame segment_94 after this interpretation is:

```
frame : segment_94.
slots : [
  [location,"value",[[63,143,151],[64,142,152],[65,141,151],[66,141,151],
    [67,141,151],[68,141,151],[69,141,151],[70,141,151],[71,141,151],
    [72,141,151],[73,142,151],[74,142,151],[75,142,151],[76,142,151],
    [77,142,151],[78,141,151],[79,141,151],[80,141,151],[81,141,150],
    [82,141,150],[83,141,150],[84,141,151],[85,141,151],[86,141,151],
    [87,141,150],[88,141,151],[89,141,151],[90,141,150],[91,141,151],
    [92,141,151],[93,141,150],[94,141,150],[95,143,149]]],
  [shape_1,"value",2.784E1],
  [window,"value",[63,141,95,152]],
  [size,"value",345],
  [mean_ch_1,"value",8.665E1],
  [mean_ch_2,"value",3.671E1],
  [mean_ch_3,"value",4.123E1],
  [mean_ch_4,"value",4.723E1],
  [mean_ch_5,"value",4.723E1],
  [mean_ch_6,"value",5.934E1],
  [max_ch_1,"value",91],
  [max_ch_2,"value",43],
  [max_ch_3,"value",45],
  [max_ch_4,"value",93],
  [max_ch_5,"value",109],
  [max_ch_6,"value",73],
  [min_ch_1,"value",80],
  [min_ch_2,"value",32],
  [min_ch_3,"value",34],
  [min_ch_4,"value",31],
  [min_ch_5,"value",74],
  [min_ch_6,"value",33],
  [ako,"if_needed,compute_ako],
  [segments_linked,"if_needed,compute_segments_linked],
  [classify_segments,"if_needed,compute_classify_segments],
  [not_mergeable_to,"value",[[segment_140,'because_of',min_max],
    [segment_129,'because_of',min_max],[segment_125,'because_of',mean],
    [segment_120,'because_of',mean],[segment_83,'because_of',min_max]]],
  [interpreted_as,"value",field],
  [interpretation,"value",bottom_up]
].
```



Segment 94 :

- Window : [63,141,95,152]

Width = 12

Length = 33

- Size : 345

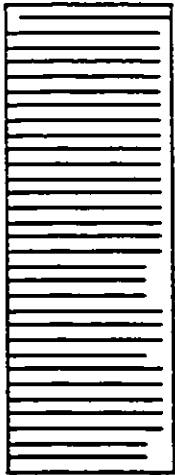
-Number of Lines filled : 33

**IF the segment width is at least 10
and the segment length is at least 10
and the segment size is at least 100
and at least (33-3) lines of the window are filled**

Fig 2.5.14

Bottom-up interpretation of a field :

Conditions



Segment 94 :

- Window : [63,141,95,152]
Width = 12
Length = 33

- Size : 345

-Number of Lines filled : 33

THEN

The bottom up interpretation of the segment is field.

Fig 2.5.15

Bottom-up interpretation of a field :

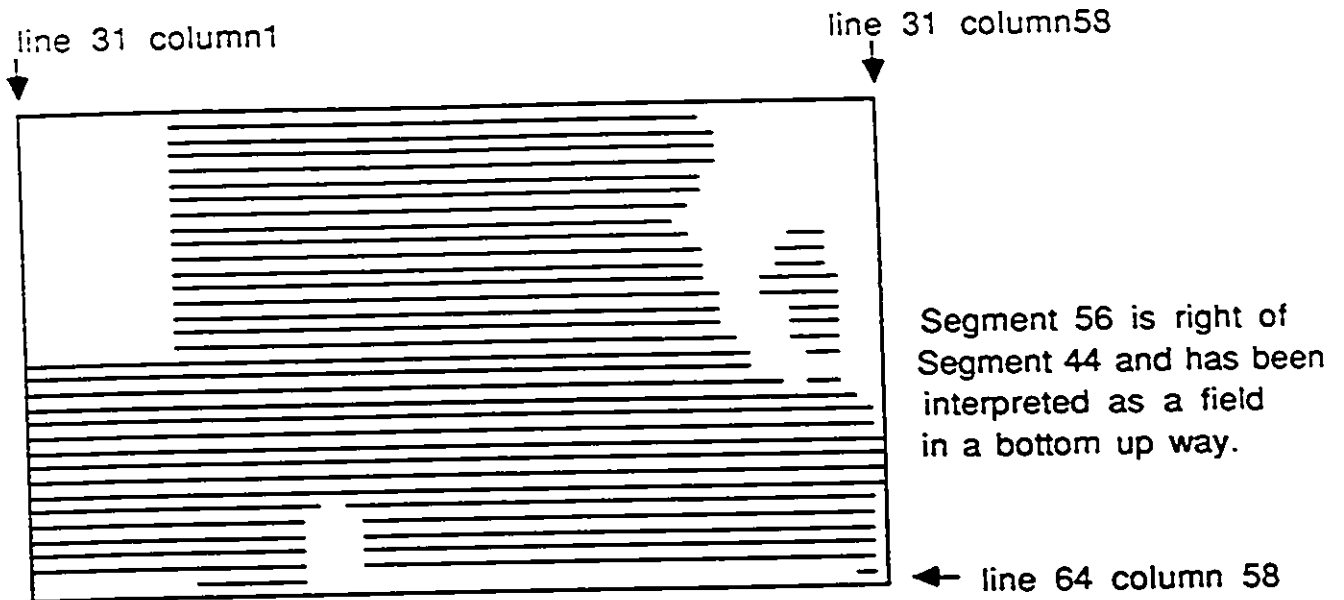
Action

If this bottom-up approach fails to give an interpretation for the current segment, LOUSTIC fires the evidence accumulation approach. Segment_44 could not be interpreted in a pure bottom-up way. LOUSTIC searches for all the surrounding segments and their interpretation. If at least two of these segments are interpreted as fields, then segment_44 which is a potential candidate for being a field due to its width, length and size, is interpreted as a field. In this particular example, both segment_56 and segment_94 have been interpreted as fields using the bottom-up approach (Fig 2.5.16 and Fig 2.5.17). Therefore, LOUSTIC concludes that segment_44 is also a field and the two interpretation slots are added to its frame.

```

frame : segment_44.
slots : [
  [interpreted as,"value",field],
  [interpretation,"value",by_evidence],
  [shape_1,"value",3.04E1],
  [size,"value",1423],
  [mean_ch_1,"value",7.624E1],
  [mean_ch_2,"value",3.133E1],
  [mean_ch_3,"value",2.701E1],
  [mean_ch_4,"value",8.769E1],
  [mean_ch_5,"value",8.769E1],
  [mean_ch_6,"value",1.998E1],
  [max_ch_1,"value",95],
  [max_ch_2,"value",46],
  [max_ch_3,"value",51],
  [max_ch_4,"value",113],
  [max_ch_5,"value",97],
  [max_ch_6,"value",50],
  [min_ch_1,"value",70],
  [min_ch_2,"value",28],
  [min_ch_3,"value",22],
  [min_ch_4,"value",49],
  [min_ch_5,"value",45],
  [min_ch_6,"value",12],
  [ako.if_needed,compute_ako],
  [segments linked,if needed,compute segments linked],
  [classify_segments,if needed,compute classify_segments],
  [not mergeable to,"value",[[segment_96,'because_of',means],
  [segment_93,'because_of',means],[segment_92,'because_of',means],
  [segment_91,'because_of',means],[segment_89,'because_of',means],
  [segment_75,'because_of',means],[segment_67,'because_of',means],
  [segment_64,'because_of',means],[segment_62,'because_of',means],
  [segment_59,'because_of',means],[segment_57,'because_of',min_max],
  [segment_55,'because_of',means],[segment_49,'because_of',means]]],
  [window,"value",[31,1,64,58]],
  [location,"value",[[31,41,42],[32,11,46],[33,11,47],[34,11,47],
  [35,11,47],[36,11,46],[37,11,46],[38,11,45],[39,11,44],[40,11,45],
  [40,52,54],[41,11,46],[41,51,54],[42,11,46],[42,51,54],[43,11,46],
  [43,50,55],[44,11,47],[44,50,55],[45,11,47],[45,52,55],[46,11,47],
  [46,53,55],[47,11,48],[47,52,55],[48,1,49],[48,53,55],[49,1,49],
  [49,54,54],[50,1,51],[50,53,55],[51,1,56],[52,1,57],[53,1,57],
  [54,1,58],[55,1,58],[56,1,58],[57,1,58],[58,1,20],[58,22,57],
  [59,1,19],[59,23,57],[60,1,19],[60,23,57],[61,1,19],[61,23,57],
  [62,1,19],[62,23,57],[63,12,19],[63,30,42],[63,56,57],[64,31,38]]],
  [merged with,"value",[[segment_90,'at_step',3],[segment_66,'at_step',4],
  [segment_63,'at_step',5],[segment_60,'at_step',6],
  [segment_54,'at_step',7]]]
].

```

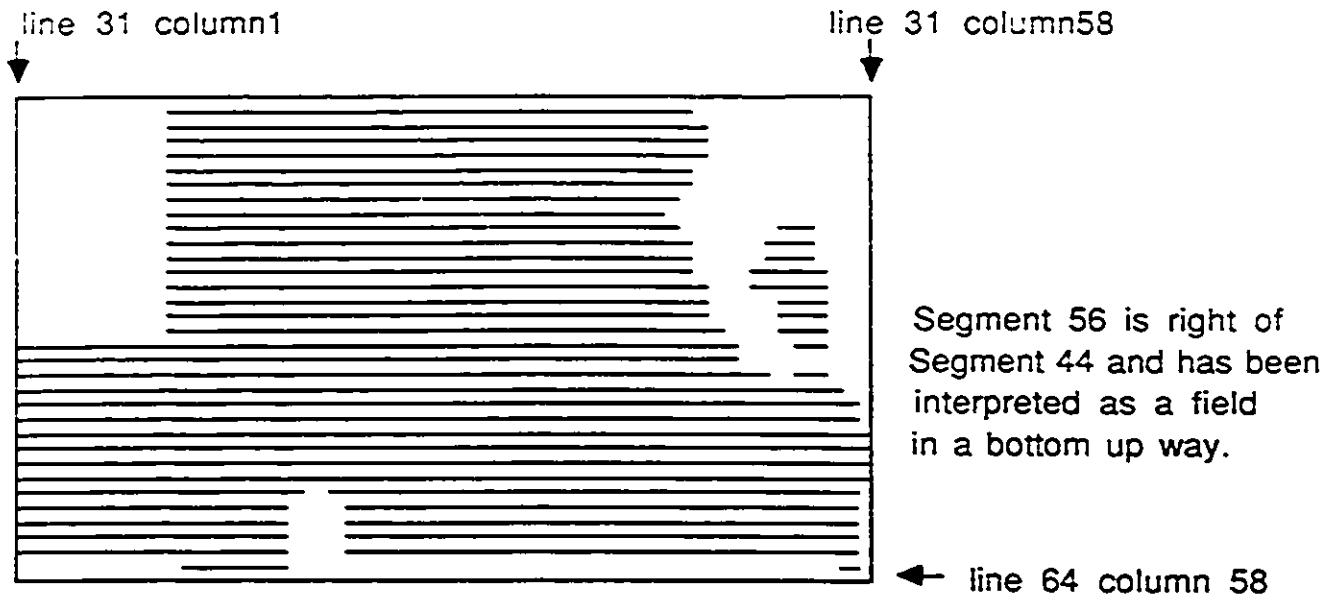


Segment 96 is right of and under Segment 44 and has been interpreted as a field in a bottom up way

- IF Segment width is at least 10 (Width = 34)**
- and Segment length is at least 10 (Length = 59)**
- and Segment size is at least 100 (Size 1423)**
- and at least 2 near by Segments have been interpreted as fields**

Fig 2.5.16

Interpretation of a field using evidence accumulation :
Conditions



Segment 96 is right of and under Segment 44
and has been interpreted as a field in a bottom up way

THEN

the interpretation of the Segment is field.

Fig 2.5.17

Interpretation of a field using evidence accumulation :

Action

Suppose a failure occurs in this process. The user has then the opportunity to resume the evidence accumulation approach. LOUSTIC prompts the user for a coefficient which corresponds to the number of pixels the user wants to expand the window of the current segment. If this coefficient is 3, then LOUSTIC expands the window of the current segment by three pixels and starts its evidence accumulation approach on this new window.

Finally, another failure in this process fires the top-down approach. The user is sent back to the LLKAS and may change the parameters of the merging and splitting rule. LOUSTIC then resumes its entire analysis on the segment using these new coefficients.

2.6 Implementation

LOUSTIC is designed to use the CCRS remote sensing shell called RESHELL.

RESHELL is a hierarchically designed system that is controlled by a knowledge engineer with meta and object rules supplied [Goodenough et al 87][Reshell85,87]. RESHELL includes all the typical components of an expert system support environment as described in Fig 1.1.4, but the most important ones for LOUSTIC are:

- the man-machine interface and the trapped commands. These commands allow interaction between the user and the system at any stage of the process;
- the Prolog Engineering Interface used for creating, debugging and testing M-Prolog code;
- the Frame Engineering Interface used for editing, displaying and testing of frames and frame daemons.

More information on the internal structure of RESHELL is available in [Goodenough et al 87].

LOUSTIC is organized into a set of M-Prolog modules, M-Prolog is the Logicware's version of Prolog. Each module contains M-Prolog facts and predicates and performs only one particular application. The principal modules are:

- The Interface User Module includes the trapped commands and man-machine interaction capabilities.
- The System Module includes the control rules as well as the strategy rules.
- The Merge Module includes the merging rules and associated processes.
- The Split Module includes the splitting rules and associated processes.
- The Interpret Module includes the high level rules and interpretation processes.

LOUSTIC is to run on the CCRS Landsat Digital Image Analysis System (LDIAS) Vax computers and AI Vax workstations using the VMS operating system [Goodenough87].

2.7 Results:

2.7.1 Original Image:

The image processed by the prototype system is a 200 * 200 window of a Landsat thematic mapper picture (Fig 2.7.1 and Fig 2.7.2). Six channels out of seven are used, the thermal band being omitted, and the intensities on each channel are in the range 0 to 255. The samplings of the LANDSAT Thematic Mapper pictures is 25 meters. The scene analyzed contains among other features fields, roads, an airport and lakes.

A human interpretation of this original image results in the identification of about 40 regions which can be interpreted as fields.

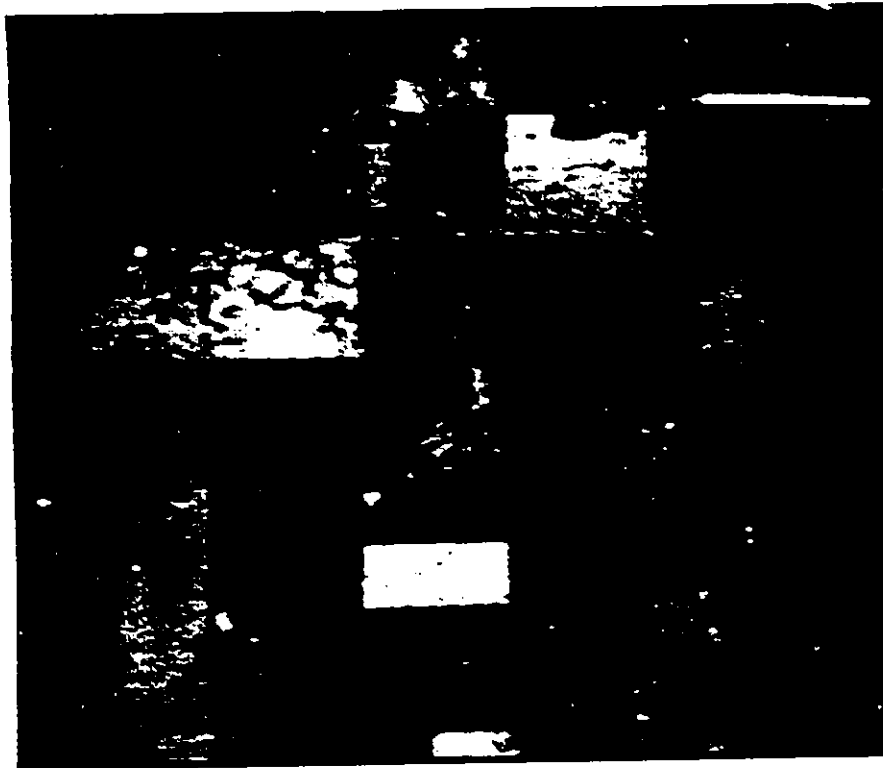


Fig 2.7.1.1

Original LANDSAT Image

Channels 1, 2, 3



Fig 2.7.1.2
Original Picture
Channels 1,4,6

2.7.2 Segmented picture and low level routines:

This original image is then processed by the VLLKAS. The segmented picture, output of the SEGMNT program is depicted in Fig 2.7.2. During the SEGMNT process, the user is prompted for a smoothing parameter which acts as a filter. In the case of the prototype image, the greater the smoothing parameter, the less segments generated. Besides, The UNITRU program which generates the spatial attributes for the segment asks for an input with a maximum of 256 segments. A smoothing parameter of 75 was selected and SEGMNT generated 254 segments. These 254 segments were then translated into frames to form the VLLFB (Very Low Level Frame Base).

If we attempt to compare the segmented picture with the original image, we can conclude that the segmentation algorithm did not provide the interpretation system with all the areas of interests we may have expected. Some regions, such as the large grey one at the middle were the results of an "undersegmentation", that is the smoothing parameter was too weak. On the other hand, many small segments were the results of an "oversegmentation" like the small segments at the top of the segmented picture. One of the objectives of LOUSTIC is then to use the general knowledge it has on segmentation to split all the segments which satisfy the conditions of the splitting rule, and to merge the ones which satisfy the conditions of the merging rules. Having done this analysis, LOUSTIC then tries to interpret the large segments selected as possible candidates for being fields.



Fig 2.7.2

Segmented Picture

2.7.3 Experiments:

In a first set of experiments, a segment was seen as small if its size was less than 10 pixels. The HLRB (High Level Rule Base) contains knowledge on fields and not_a_field. That is, a potential candidate for field was at least 10 pixels long, 10 pixels large and with a size of at least 100 pixels. A segment was classified in a bottom-up way as not_a_field if it did not match the previous conditions.

With this knowledge, we obtained the following results (Fig 2.7.3.1):

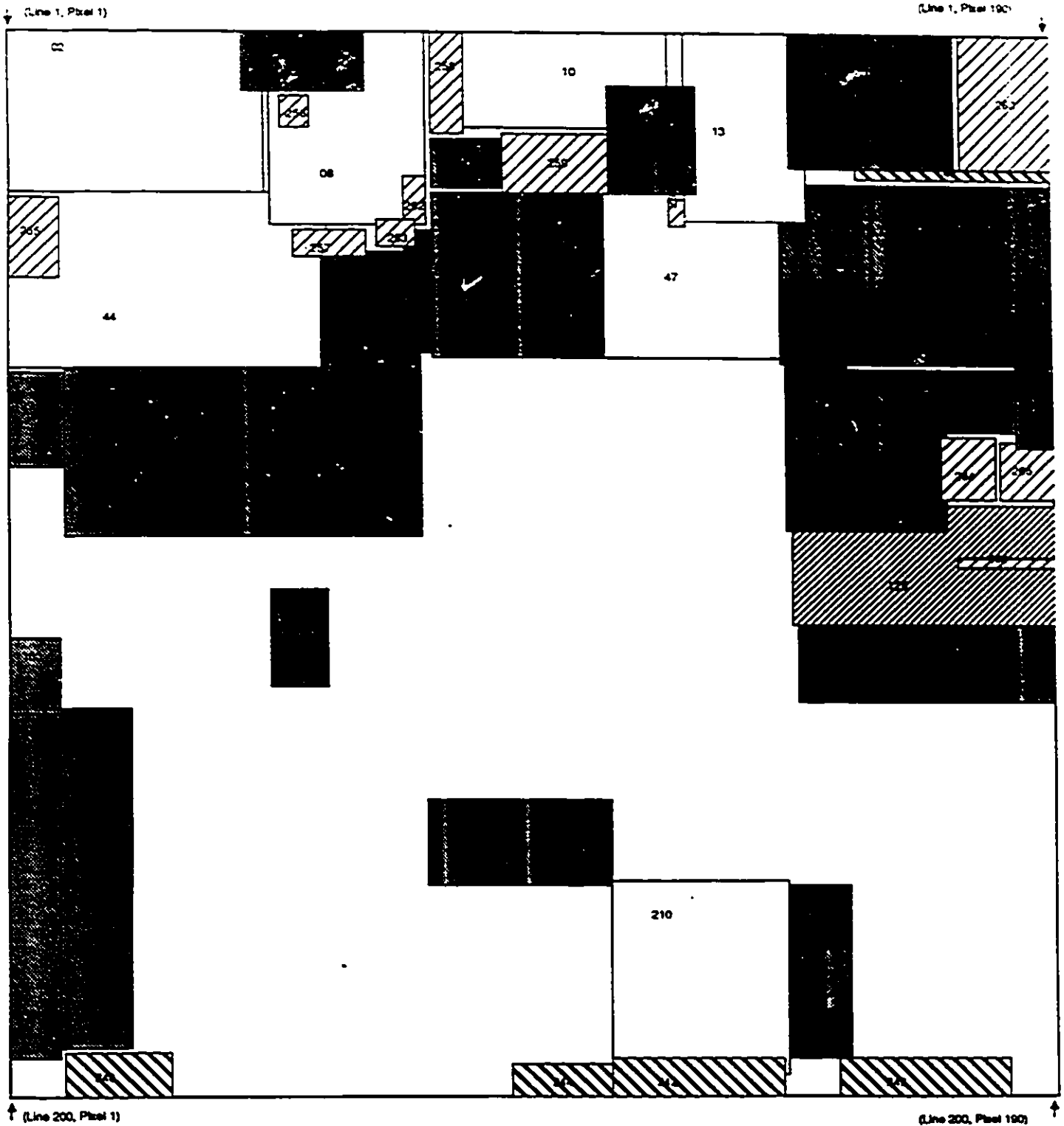
Of the 254 segments generated by the initial segmentation:

- LOUSTIC selected 34 of these segments as potential candidates for fields according to the knowledge it had on fields. That is, 34 segments in the frame segments database had a size of at least 100 pixels.
- The knowledge of LOUSTIC on general segmentation enabled it to split 12 of these 34 segments , thus creating 12 new frame-segments . We did not attempt to later analyze nor interpret these 12 new segments.
- 18 segments of the initial 254 were merged to potential candidates for fields as they matched the merging rule criterion.
- The knowledge LOUSTIC had on fields applied on the 34 segments selected as potential candidates for fields gave the following results:
 - 20 of these potential candidates for field were interpreted as a field using the initial bottom-up approach.
 - 1 potential candidate was interpreted as a field with the evidence accumulation approach. This segment corresponded to a region large and wide enough to be interpreted as a field but which presented large discontinuities : its location did not fill its minimal window. However, it was surrounded by three other segments previously interpreted as fields and was thus declared to be a field.

- 5 of these potential candidates were interpreted as not_a_field as they did not satisfy one of the conditions of the rule for fields. Among these 5 segments is 1 airport runway and 4 parts of real fields. However these fields were cut in some parts by the limits chosen for the image processing window. The system would successfully recognize them if it had had the entire image.

- LOUSTIC was unable to interpret the remaining 8 potential candidates segments. 2 of them would require further knowledge as they correspond to a lake and bushes. 1 of them would be successfully recognized using the evidence accumulation approach after a second path as it is surrounded by at least 3 other segments interpreted as fields. The last five uninterpreted segments are the results of the initial segmentation which provided some large "undersegmented" regions such as the one located in the middle of the segmented picture (Fig 2.7.2).

Thus, out of 40 initial fields as discovered by a human segmentation of the original image, LOUSTIC successfully recognized 21 of them. 3 more may have been recognized if we had attempted to interpret the new segments generated by the splitting rule. The last 16 are for the most part located in the large "undersegmented" region in the middle of the segmented picture. They were not detected because of the initial segmentation which did not provide enough segments.








-  New segments
-  Segments interpreted as fields using the evidence accumulation approach
-  Segments interpreted as not_a_field
-  Segments interpreted as fields in a bottom-up way
-  Uninterpreted segments

Fig 2.7.3.1
Interpreted segments

This first set of experiments was performed through an automatic process. That is, LOUSTIC selected all the segments of at least 100 pixels as potential candidates for fields. It then processed each of these candidates starting with the first of the list. In this example LOUSTIC began its analysis with segment_02 and stopped at segment_244. A second interactive path may benefit from these conclusions to interpret new segments with the evidence accumulation approach. For example, during the automatic path, segment_44 was processed before segment_56 and segment_96, and LOUSTIC was able to interpret it. However, LOUSTIC then interpreted both segment_56 and segment_96 as fields in a bottom-up manner. Thus a second interpretation path on segment_44 will lead to its interpretation as a field using the evidence accumulation approach.

One of the objectives of this research is to build a flexible CVS for the segmentation and interpretation of images. LOUSTIC identified some fields in the picture. Provided the network of fields, LOUSTIC may be able to identify roads or portions of roads at the intersection of fields. As seen in the segmented picture, the segmentation algorithm is not able to detect long, straight segments matching the roads. Instead, it detects part of roads of approximately 2 pixels wide and 3 to 10 pixels long (50 meters by 75 meters). Thus, in a second set of experiments, we added a rule in order to detect these portions of roads in a bottom-up way. The strategy is then to identify all the segments at the intersection of segments interpreted as fields and apply this new rule.

The rule is as follows :

IF width of segment is at least 2

and length of segment is at least 3

and size of segment is at least 6

THEN the interpretation of segment is road in a bottom-up way.

Segment_88 is such a segment. It has been classified by LOUSTIC as "above" segment_96 and "under" segment_44. Both segment_96 and segment_44 are fields according to LOUSTIC. Thus segment_88 is interpreted as a portion of a road (Fig 2.7.3.3 and Fig 2.7.3.4).

A possible improvement would be to classify segment as portion of roads if in a close neighborhood of this segment, similar segments have been classified as portions of road. LOUSTIC may then identify portion of the road network.

Segment 44 is above
and interpreted as a field



Segment 96 is under
and interpreted as a field

Segment 89 :

-Window : [63,1,64,11]

Width = 2

Length = 11

-Size : 22

-Number of Lines filled : 2

IE

The segment width is at least 2
and the segment length is at least 3
and the segment size is at least 6
and the lines of the window are filled

Fig 2.7.3.3

Bottom-up interpretation of a road :

Conditions

Segment 44 is above
and interpreted as a field



Segment 96 is under
and interpreted as a field

Segment 89 :

-Window : [63.1.64.11]

Width = 2

Length = 11

-Size : 22

-Number of Lines filled : 2

THEN

The bottom up interpretation of the segment is road.

Fig 2.7.3.4

Bottom-up interpretation of a road :

Action

2.8 Summary and conclusions

We developed a Computer Vision System for the segmentation and analysis of remotely sensed imagery. This system involves a powerful knowledge representation scheme using frames to represent image segments and a control structure incorporating both bottom-up and top-down approaches to recognize some objects in a satellite picture. We proved that a limited set of segmentation rules could refine a low level segmented picture, the introduction of high level rules of the world under analysis could then help detect and identify different kind of objects present in the scene.

The following conclusions can be drawn from the work performed thus far:

- The results so far are very favourable for performing segmentation/interpretation using an artificial intelligence knowledge based system. Using a limited set of rules, the system was able to identify the majority of the fields present in the scene.
- The aim of this research was to build a CVS for the segmentation and analysis of Landsat images and LOUSTIC is the first prototype system. However LOUSTIC is still a system for segmentation, but we proved that the frame representation scheme is a uniform, straightforward way of combining information from multiple sources of knowledge.
- The first extension of the existing system is to add additional information for the segmentation process. The high level segmentation rules can be improved with the addition of knowledge on texture, depth or shape. Adding other kinds of knowledge is just a matter of adding slots in the frames . Similarly, LOUSTIC could benefit from the analysis and conclusions of other existing systems. The flexibility of this frame representation enables the addition of other kinds of information.

- Although LOUSTIC uses knowledge on segmentation to refine the low level process, it still relies on an acceptable segmented picture. To overcome this problem, Matsuyama used a top-down segmentation scheme where the system uses its knowledge of low level segmentation to apply the best possible algorithm. This strategy could be implemented in LOUSTIC. For example the best channel to identify hydrography elements is channel 5. LOUSTIC could use this knowledge to fire a segmentation process that finds large segments in channel 5 and then tries to detect lakes.

- LOUSTIC has no constraints in terms of the number of segments in the frame database. However the limitation of the number of segments for the UNITRU FORTRAN routine imposes a segmented picture with some large "undersegmented " region as seen in Fig 2.7.2. Further research and improvements on this low level routine will greatly improve the quality of the results.

- The selection of the segment to analyze still depends on the user. A more powerful selection should give better results. For example, given the interpretation of a segment, LOUSTIC could then analyze and interpret the surrounding segments. The detection of another field could fire an interpretation process which identifies roads or portion of roads at the intersection of fields and so on.

- The control process as implemented in LOUSTIC uses both top-down and bottom-up interpretation. However, the way they are implemented, LOUSTIC always starts its interpretation using the bottom-up approach followed by the top-down approach. A more powerful scheme making use of the context and the objects interpreted so far will improve the interpretation stage.

- Similarly, more intensive use could be made with the AKO (a kind of) link. Using this link, implemented as a slot of a frame segment, a segment interpreted as a field could inherit all the knowledge of a frame field. This knowledge could fire other processes to refine or confirm the interpretation.
- A trapped command allowing display of segments could be implemented in the system.
- The picture processed represents fields in Saskatchewan, a province where the fields are large. To prove the effectiveness of LOUSTIC, we still have to test it on other pictures .

Each of these modifications is aimed at increasing the accuracy and utility of the segmentation and/or the interpretation processes. They would not involve any major modifications to the system, due to the flexibility of the RESHELL environment and the LOUSTIC environment (representation of segments using frames, implementation of both bottom-up and top-down approach, interaction capabilities). However, these modifications would greatly enhance the reliability of the results.

3.1 REFERENCES

1. [Basden84] Basden, A.; "On the Application of Expert Systems"; *Developments in Expert Systems*, Coombs, M.J. (ed); 1984; pp 59-75.
2. [CVS78] Hanson, A.R., Riseman, E.M.; "*Computer Vision Systems*"; Academic Press; 1978.
3. [Duda79] Duda, R.O., Hart, P.E., Konolige, K., Reboh, R.; "A Computer-based Consultant for Mineral Exploration"; *Technical Report SRI International*, Sept 1979.
4. [Erman et al80] Erman, L.D., Hayes-Roth, F., Lesser, V.R., Reddy, D.R.; "The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty"; in *Computing Surveys*, Vol 12, No. 2; June 1980, pp213-253.
5. [Gevarter87] Gevarter, W.B.; "The Nature and Evaluation of Commercial Expert System Building Tools"; in *Computer*, May 1987, pp 24-41.
6. [Gilmore84] Gilmore, J.F.; "Artificial Intelligence in Automatic Target Recognition"; in *SPIE Vol 504, Applications of Digital Image Processing VII*; 1984, pp 33-39.
7. [Goodenough87] Goodenough, D.G.; "The integration of remote sensing and geographic information systems"; in *Proc. ISPRS Commission VII*; Enschede, Netherlands (in Press); 1987.

8. [Goodenough et al 87] Goodenough, D.G., Goldberg,M., Plunkett,G., Zelek,J.; "An Expert System for Remote Sensing"; in *IEEE trans. on Geoscience and Remote Sensing*, Vol GE-25 No3; 1987, pp 349-359.
9. [Hanson78] Hanson,A.R., Riseman, E.M.; "Segmentation of Natural Scenes"; in *Computer Vision Systems*, Academic Press; 1978, pp 129-163.
10. [Hanson and Riseman78] Hanson, A.R., Riseman, E.M. ; "VISIONS: a Computer System for Interpreting Scenes"; in *Computer Vision Systems*, Academic Press; 1978, pp 129-163.
11. [Levine84] Nazif, A.M., Levine, M.D.; "Low Level Image Segmentation: an Expert System"; in *IEEE trans. on Pattern Analysis and Machine Intelligence*, Vol PAMI-6 No5; 1984, pp555-577.
12. [MacKeown85] Mc Keown, D.M., Wilson, A. H., Mc Dermott, J. ; " Rule-based Interpretation of Aerial Imagery"; in *IEEE trans. on Pattern Analysis and Machine Intelligence*, Vol PAMI-7 No5; Sept 1985 pp 570-585.
13. [MacKeown86] McKeown, D.M. ; "The Role of AI in the Integration of Remotely Sensed Data with Geographic Information Systems"; *IAPR TC 7 Workshop on Analytical Methods in Remote Sensing for GIS*, Paris, France; October1986.

14. [Matsuyama86] Hwang, V.S., Davis, L.S., Matsuyama, T.; "The Hypothesis Integration in Image Understanding Systems "; in *Computer Graphics and Image Processing*, Dec 1986, pp 321-371.
15. [Matsuyama86] Matsuyama, T.; "Present and Future Role of Artificial Intelligence in Remote Sensing: Knowledge-Based Aerial Image Understanding Systems and Expert Systems for Image Processing"; *IAPR TC 7 Workshop on Analytical Methods in Remote Sensing for GIS*, Paris, France; October 1986, pp 223-258.
16. [Mooneyhan83] Mooneyhan, D.W.; "The Potential of Expert Systems for Remote Sensing Applications"; *Proceedings 1983 International Geoscience and Remote Sensing Symposium*; 1983 pp 4.1-4.5.
17. [Plunkett86] Plunkett, G.W. ; University of Ottawa, MASC thesis; 1986.
18. [Reshell85] Goldberg, M., Goodenough, D.G., Alvo, M., Karam, G.; "A Hierarchical Expert System for Updating Forestry Maps with Landsat Data"; *Proceedings of the IEEE*, Vol 73, No6; June 1985, pp1045-1063.
19. [Reshell87] Karam, G.M., Fung, K.B.; "The RESHELL beginner's guide"; *Technical Report AIRP007-87 CCRS*; March 1987.
20. [Reshell87] Karam, G.M.; "Knowledge Engineering with RESHELL"; *Technical Report LOUATXZ 5-87-00.0 CCRS*; March 1987.
21. [Rich83] Rich, E.; *Artificial Intelligence*; McGraw Hill; 1983.

22. [Shortliffe76] Shortliffe,E.; *"Computer-based Medical Consultation: MYCIN"*; Elsevier; 1976.
23. [Waterman86] Waterman, D.A.; *"A Guide to Expert Systems"*; Addison-Wesley; 1986 chapter1.
25. [Winston84] Winston, P.H.; *"Artificial Intelligence"*; 2nd Edition, Addison-Wesley; 1984. chapter1.

Appendix: Typical Example Session

LOADING : es_controller - 87/07/31 11:09:39

```
* Initializing System *
* Executing RESHELL system startup file *
* RESHELL system startup file executed *
* Executing user startup file *
* User startup file executed *
* Loading help file *
* Help file loaded *
* Loading frames database *
* Frames database loaded *
* Loading frame daemons *
* Frames daemons loaded *
* Loading knowledge and context for: projet *
* All knowledge loaded *
* Initialized system *
```

*** User Interface ***

Enter one of run resume save quit exit help {help}

pei

Warning, no expert in focus

*** Prolog Engineering Interface (level 1) ***

Enter one of load save add replace delete display interpreter bye help
{interpreter}

interpreter

* MProlog Interpreter Invoked *

start_the_system.

How many segments in your picture ?

254

You are in the low level KAS

Knowledge acquisition for :

Enter one of merge split bye {bye}

merge

minimum size of a segment seen as small

{15}

10

minimum difference of the means to merge a small segment

{15}

15

the merging rules with your choice are:

IF segment size is less than 10

and if the means of this segment

are in the min-max range of the big segment

THEN merge the two segments

IF segment size is greater or equal to 10

and if the difference of the means on the 6 channels

is less than 15

THEN merge the two segments

Knowledge acquisition for :
Enter one of merge split bye {bye}
split
A segment will be split into two parts
IF the length of the window is greater than 10
and if 2 the length of the line
does not fit in the window

and if the length of the window of
the new possible segment is greater than 5

OK with this choice
Enter one of yes no {yes}
yes
Knowledge acquisition for :
Enter one of merge split bye {bye}
bye
You are in the high level KAS
Precision for a line to fit in the window
{3}

3
Knowledge acquisition for :
Enter one of field road bye {bye}
field
Minimum length of a field
{10}

10
Minimum width of a field
{10}

10
A segment will be interpreted in a bottom up way as a field
IF segment_width is at least 10
and segment_length is at least 10
and segment_size is at least 100

Knowledge acquisition for :
Enter one of field road bye {bye}
road
Minimum length of a road
{4}

3
Minimum width of a road
{2}

2
A segment will be interpreted in a bottom up way as a road
IF segment_width is at least 2
and segment_length is at least 3
and segment_size is at least 6

Knowledge acquisition for :
Enter one of field road bye {bye}
bye

What do you want to do ?
 Enter one of focus_all_segments focus_big_segments analyze_all_segments
 quit {qu:t}
 analyze_all_segments

segment_02 split into two segments
 New segment created at :{32,1,47,10}

impossible merge between segment_02 and segment_44
 the min-max are OK, I may merge segment_43 to segment_02
 the min-max are OK, I may merge segment_41 to segment_02
 the min-max are OK, I may merge segment_38 to segment_02
 the min-max are OK, I may merge segment_32 to segment_02
 impossible merge between segment_02 and segment_29
 impossible merge between segment_02 and segment_28
 impossible merge between segment_02 and segment_24
 impossible merge between segment_02 and segment_22
 impossible merge between segment_02 and segment_21
 impossible merge between segment_02 and segment_20
 impossible merge between segment_02 and segment_19
 impossible merge between segment_02 and segment_07
 impossible merge between segment_02 and segment_06
 impossible merge between segment_02 and segment_05
 impossible merge between segment_02 and segment_04
 the min-max are OK, I may merge segment_03 to segment_02
 impossible merge between segment_02 and segment_01
 segment_38 merged to segment_02
 segment_32 merged to segment_02
 segment_03 merged to segment_02
 unable to interpret segment_02

some statistics :{segment_02,{31,lines_in_window},{22,lines_full}}

segment_07 split into two segments
 New segment created at :{13,50,19,55}

impossible merge between segment_07 and segment_22
 impossible merge between segment_07 and segment_21
 impossible merge between segment_07 and segment_08
 impossible merge between segment_07 and segment_06
 impossible merge between segment_07 and segment_02
 segment_07 interpreted as field
 Interpretation is bottom_up

some statistics :{segment_07,{12,lines_in_window},{12,lines_full}}

segment_08 split into two segments
 New segment created at :{38,52,43,65}

impossible merge between segment_08 and segment_256
 impossible merge between segment_08 and segment_56
 impossible merge between segment_08 and segment_55
 impossible merge between segment_08 and segment_50
 impossible merge between segment_08 and segment_49
 impossible merge between segment_08 and segment_48
 the means are OK, I may merge segment_44 to segment_08
 the min-max are OK, I may merge segment_43 to segment_08
 impossible merge between segment_08 and segment_42

the min-max are OK, I may merge segment_41 to segment_08
 impossible merge between segment_08 and segment_36
 impossible merge between segment_08 and segment_33
 impossible merge between segment_08 and segment_29
 impossible merge between segment_08 and segment_07
 unable to interpret segment_08

some statistics :[segment_08,[37,lines_in_window],[32,lines_full]]

segment_09 split into two segments
 New segment created at :[1,77,20,83]

impossible merge between segment_09 and segment_42
 impossible merge between segment_09 and segment_10
 segment_09 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_09,[10,lines_in_window],[10,lines_full]]

segment_10 split into two segments
 New segment created at :[20,90,31,109]

impossible merge between segment_10 and segment_35
 impossible merge between segment_10 and segment_34
 impossible merge between segment_10 and segment_31
 impossible merge between segment_10 and segment_30
 impossible merge between segment_10 and segment_27
 impossible merge between segment_10 and segment_26
 impossible merge between segment_10 and segment_23
 impossible merge between segment_10 and segment_12
 impossible merge between segment_10 and segment_11
 unable to interpret segment_10

some statistics :[segment_10,[19,lines_in_window],[16,lines_full]]

impossible merge between segment_13 and segment_53
 impossible merge between segment_13 and segment_52
 impossible merge between segment_13 and segment_51
 impossible merge between segment_13 and segment_47
 impossible merge between segment_13 and segment_26
 impossible merge between segment_13 and segment_25
 impossible merge between segment_13 and segment_18
 impossible merge between segment_13 and segment_14
 impossible merge between segment_13 and segment_12
 unable to interpret segment_13

some statistics :[segment_13,[36,lines_in_window],[33,lines_full]]

impossible merge between segment_14 and segment_40
 impossible merge between segment_14 and segment_39
 impossible merge between segment_14 and segment_25
 impossible merge between segment_14 and segment_18
 impossible merge between segment_14 and segment_13
 segment_14 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_14,[26,lines_in_window],[25,lines_full]]

segment_18 split into two segments
 New segment created at :[1,174,28,200]

impossible merge between segment_18 and segment_95
 impossible merge between segment_18 and segment_94
 the min-max are OK, I may merge segment_88 to segment_18
 impossible merge between segment_18 and segment_87
 the min-max are OK, I may merge segment_86 to segment_18
 the min-max are OK, I may merge segment_83 to segment_18
 impossible merge between segment_18 and segment_82
 the min-max are OK, I may merge segment_79 to segment_18
 impossible merge between segment_18 and segment_78
 impossible merge between segment_18 and segment_77
 impossible merge between segment_18 and segment_73
 impossible merge between segment_18 and segment_71
 impossible merge between segment_18 and segment_70
 the min-max are OK, I may merge segment_69 to segment_18
 the min-max are OK, I may merge segment_68 to segment_18
 the min-max are OK, I may merge segment_65 to segment_18
 impossible merge between segment_18 and segment_47
 impossible merge between segment_18 and segment_13
 segment_88 merged to segment_18
 segment_79 merged to segment_18
 segment_69 merged to segment_18
 segment_68 merged to segment_18
 segment_65 merged to segment_18
 segment_18 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_18,[35,lines_in_window],[34,lines_full]]

segment_26 split into two segments
 New segment created at :[32,120,37,123]

impossible merge between segment_26 and segment_47
 impossible merge between segment_26 and segment_35
 impossible merge between segment_26 and segment_31
 impossible merge between segment_26 and segment_30
 the means are OK, I may merge segment_27 to segment_26
 impossible merge between segment_26 and segment_13
 impossible merge between segment_26 and segment_12
 impossible merge between segment_26 and segment_10
 segment_26 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_26,[21,lines_in_window],[20,lines_full]]

impossible merge between segment_40 and segment_260
 segment_40 interpreted as not_a_field
 Interpretation is sure

some statistics :[segment_40,[3,lines_in_window],[3,lines_full]]

segment_42 split into two segments
 New segment created at :{28,72,37,76}

impossible merge between segment_42 and segment_45
 segment_42 interpreted as field
 Interpretation is bottom_up

some statistics :{segment_42,[24,lines_in_window],[24,lines_full]}

the means are OK, I may merge segment_257 to segment_44
 impossible merge between segment_44 and segment_255
 impossible merge between segment_44 and segment_96
 impossible merge between segment_44 and segment_93
 impossible merge between segment_44 and segment_92
 impossible merge between segment_44 and segment_91
 the min-max are OK, I may merge segment_90 to segment_44
 impossible merge between segment_44 and segment_89
 impossible merge between segment_44 and segment_86
 impossible merge between segment_44 and segment_75
 impossible merge between segment_44 and segment_67
 the min-max are OK, I may merge segment_66 to segment_44
 impossible merge between segment_44 and segment_64
 the means are OK, I may merge segment_63 to segment_44
 impossible merge between segment_44 and segment_62
 the min-max are OK, I may merge segment_60 to segment_44
 impossible merge between segment_44 and segment_59
 impossible merge between segment_44 and segment_57
 the means are OK, I may merge segment_56 to segment_44
 impossible merge between segment_44 and segment_55
 the min-max are OK, I may merge segment_54 to segment_44
 impossible merge between segment_44 and segment_49
 the min-max are OK, I may merge segment_43 to segment_44
 the means are OK, I may merge segment_08 to segment_44
 segment_90 merged to segment_44
 segment_66 merged to segment_44
 segment_63 merged to segment_44
 segment_60 merged to segment_44
 segment_54 merged to segment_44
 unable to interpret segment_44

some statistics :{segment_44,[34,lines_in_window],[15,lines_full]}

impossible merge between segment_45 and segment_86
 the min-max are OK, I may merge segment_85 to segment_45
 impossible merge between segment_45 and segment_82
 impossible merge between segment_45 and segment_76
 impossible merge between segment_45 and segment_74
 impossible merge between segment_45 and segment_46
 impossible merge between segment_45 and segment_42
 segment_45 interpreted as field
 Interpretation is bottom_up

some statistics :{segment_45,[32,lines_in_window],[32,lines_full]}

impossible merge between segment_47 and segment_261
 impossible merge between segment_47 and segment_86
 impossible merge between segment_47 and segment_83
 impossible merge between segment_47 and segment_30
 impossible merge between segment_47 and segment_53
 impossible merge between segment_47 and segment_52
 impossible merge between segment_47 and segment_51
 impossible merge between segment_47 and segment_18
 impossible merge between segment_47 and segment_13
 unable to interpret segment_47

some statistics :[segment_47,[32,lines_in_window],[28,lines_full]]

segment_56 split into two segments
 New segment created at :[36,67,41,74]

impossible merge between segment_56 and segment_257
 impossible merge between segment_56 and segment_100
 impossible merge between segment_56 and segment_96
 impossible merge between segment_56 and segment_86
 impossible merge between segment_56 and segment_72
 impossible merge between segment_56 and segment_67
 impossible merge between segment_56 and segment_62
 the means are OK, I may merge segment_44 to segment_56
 segment_56 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_56,[24,lines_in_window],[23,lines_full]]

impossible merge between segment_94 and segment_140
 impossible merge between segment_94 and segment_129
 impossible merge between segment_94 and segment_125
 impossible merge between segment_94 and segment_120
 impossible merge between segment_94 and segment_86
 impossible merge between segment_94 and segment_83
 segment_94 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_94,[33,lines_in_window],[33,lines_full]]

impossible merge between segment_95 and segment_125
 impossible merge between segment_95 and segment_124
 impossible merge between segment_95 and segment_111
 impossible merge between segment_95 and segment_108
 impossible merge between segment_95 and segment_86
 segment_95 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_95,[17,lines_in_window],[16,lines_full]]

impossible merge between segment_96 and segment_139
 impossible merge between segment_96 and segment_136
 impossible merge between segment_96 and segment_135
 impossible merge between segment_96 and segment_131
 the min-max are OK, I may merge segment_126 to segment_96
 impossible merge between segment_96 and segment_118
 impossible merge between segment_96 and segment_115
 impossible merge between segment_96 and segment_114
 impossible merge between segment_96 and segment_113
 impossible merge between segment_96 and segment_107
 impossible merge between segment_96 and segment_103
 the means are OK, I may merge segment_100 to segment_96
 impossible merge between segment_96 and segment_92
 impossible merge between segment_96 and segment_91
 impossible merge between segment_96 and segment_86
 impossible merge between segment_96 and segment_56
 segment_126 merged to segment_96
 segment_100 merged to segment_96
 segment_96 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_96,[33,lines_in_window],[31,lines_full]]

segment_98 split into two segments
 New segment created at :[77,169,89,179]

impossible merge between segment_98 and segment_122
 impossible merge between segment_98 and segment_120
 the min-max are OK, I may merge segment_119 to segment_98
 impossible merge between segment_98 and segment_112
 the min-max are OK, I may merge segment_110 to segment_98
 the min-max are OK, I may merge segment_106 to segment_98
 the min-max are OK, I may merge segment_102 to segment_98
 the min-max are OK, I may merge segment_86 to segment_98
 segment_119 merged to segment_98
 segment_110 merged to segment_98
 segment_106 merged to segment_98
 segment_102 merged to segment_98
 segment_98 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_98,[13,lines_in_window],[13,lines_full]]

impossible merge between segment_99 and segment_133
 impossible merge between segment_99 and segment_132
 segment_99 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_99,[19,lines_in_window],[19,lines_full]]

the min-max are OK, I may merge segment_140 to segment_120
 impossible merge between segment_120 and segment_129
 impossible merge between segment_120 and segment_125
 impossible merge between segment_120 and segment_123
 impossible merge between segment_120 and segment_98
 impossible merge between segment_120 and segment_94
 impossible merge between segment_120 and segment_86
 segment_140 merged to segment_120
 segment_120 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_120,[21,lines_in_window],[20,lines_full]]

segment_125 split into two segments
 New segment created at :{78,180,89,200}

the min-max are OK, I may merge segment_160 to segment_125
 impossible merge between segment_125 and segment_158
 impossible merge between segment_125 and segment_156
 impossible merge between segment_125 and segment_155
 the min-max are OK, I may merge segment_154 to segment_125
 the min-max are OK, I may merge segment_153 to segment_125
 impossible merge between segment_125 and segment_150
 impossible merge between segment_125 and segment_146
 impossible merge between segment_125 and segment_143
 the min-max are OK, I may merge segment_141 to segment_125
 the means are OK, I may merge segment_138 to segment_125
 impossible merge between segment_125 and segment_137
 impossible merge between segment_125 and segment_120
 impossible merge between segment_125 and segment_94
 impossible merge between segment_125 and segment_86
 segment_154 merged to segment_125
 segment_153 merged to segment_125
 segment_141 merged to segment_125
 segment_125 interpreted as field
 Interpretation is by_evidence

some statistics :{segment_125,[24,lines_in_window],[9,lines_full]}

impossible merge between segment_143 and segment_150
 impossible merge between segment_143 and segment_146
 impossible merge between segment_143 and segment_125
 impossible merge between segment_143 and segment_86
 segment_143 interpreted as field
 Interpretation is bottom_up

some statistics :{segment_143,[32,lines_in_window],[32,lines_full]}

segment_146 split into two segments
 New segment created at :{100,172,112,193}

impossible merge between segment_146 and segment_174
 impossible merge between segment_146 and segment_161
 impossible merge between segment_146 and segment_160
 impossible merge between segment_146 and segment_158
 impossible merge between segment_146 and segment_143
 impossible merge between segment_146 and segment_86
 segment_146 interpreted as field
 Interpretation is bottom_up

some statistics :{segment_146,[15,lines_in_window],[1,lines_full]}

impossible merge between segment_151 and segment_171
 impossible merge between segment_151 and segment_86
 segment_151 interpreted as field
 Interpretation is bottom_up

some statistics :{segment_151,[19,lines_in_window],[19,lines_full]}

impossible merge between segment_163 and segment_181
 impossible merge between segment_163 and segment_176
 impossible merge between segment_163 and segment_175
 the means are OK, I may merge segment_162 to segment_163
 impossible merge between segment_163 and segment_147
 segment_162 merged to segment_163
 segment_163 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_163,[17,lines_in_window],[17,lines_full]]

impossible merge between segment_176 and segment_136
 impossible merge between segment_176 and segment_134
 impossible merge between segment_176 and segment_185
 impossible merge between segment_176 and segment_181
 impossible merge between segment_176 and segment_175
 impossible merge between segment_176 and segment_163
 the min-max are OK, I may merge segment_86 to segment_176
 segment_176 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_176,[64,lines_in_window],[62,lines_full]]

impossible merge between segment_181 and segment_241
 impossible merge between segment_181 and segment_185
 impossible merge between segment_181 and segment_176
 impossible merge between segment_181 and segment_175
 impossible merge between segment_181 and segment_163
 segment_181 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_181,[65,lines_in_window],[65,lines_full]]

impossible merge between segment_199 and segment_86
 segment_199 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_199,[17,lines_in_window],[16,lines_full]]

impossible merge between segment_210 and segment_242
 impossible merge between segment_210 and segment_237
 impossible merge between segment_210 and segment_235
 impossible merge between segment_210 and segment_222
 impossible merge between segment_210 and segment_220
 impossible merge between segment_210 and segment_86
 unable to interpret segment_210

some statistics :[segment_210,[37,lines_in_window],[33,lines_full]]

impossible merge between segment_212 and segment_243
 impossible merge between segment_212 and segment_235
 impossible merge between segment_212 and segment_233
 impossible merge between segment_212 and segment_228
 impossible merge between segment_212 and segment_213
 impossible merge between segment_212 and segment_86
 segment_212 interpreted as field
 Interpretation is bottom_up

some statistics :[segment_212,[33,lines_in_window],[33,lines_full]]

impossible merge between segment_240 and segment_236
 impossible merge between segment_240 and segment_86
 segment_240 interpreted as not_a_field
 Interpretation is sure

some statistics :[segment_240,[9,lines_in_window],[8,lines_full]]

impossible merge between segment_242 and segment_235
 impossible merge between segment_242 and segment_210
 impossible merge between segment_242 and segment_86
 segment_242 interpreted as not_a_field
 Interpretation is sure

some statistics :[segment_242,[8,lines_in_window],[7,lines_full]]

impossible merge between segment_243 and segment_86
 segment_243 interpreted as not_a_field
 Interpretation is sure

some statistics :[segment_243,[8,lines_in_window],[8,lines_full]]

impossible merge between segment_244 and segment_86
 segment_244 interpreted as not_a_field
 Interpretation is sure

some statistics :[segment_244,[7,lines_in_window],[7,lines_full]]

What do you want to do ?

Enter one of focus_all_segments focus_big_segments analyze_all_segments
 quit {quit}
 focus_b

Which segment to focus on

Enter one of bye segment_02 segment_07 segment_08 segment_09 segment_10
 segment_13 segment_14 segment_18 segment_26 segment_40 segment_42 segment_44
 segment_45 segment_47 segment_56 segment_94 segment_95 segment_96 segment_98
 segment_99 segment_120 segment_125 segment_143 segment_146 segment_151
 segment_163 segment_176 segment_181 segment_199 segment_210 segment_212
 segment_240 segment_242 segment_243 segment_244 {bye}
 segment_44

Processing segment_44

OK what do I do

Enter one of interpret merge both consult bye {bye}

search

*** Search features environment ***

Enter one of find help bye {find}

find

search for what

Enter one of segments_linked interpreted classify_segments location greater
 less bye {bye}

segments_linked

for which segment

{segment_01}

segment_44

[[segment_257,[right_of]], [segment_255,[must be inside]], [segment_96,[
 right_of,under]], [segment_93,[must be inside]], [segment_92,[under]], [segment_
 ,[under]], [segment_89,[must be inside]], [segment_86,[right_of,under]], [
 segment_75,[must be inside]], [segment_67,[right_of]], [segment_64,[
 must be inside]], [segment_62,[must be inside]], [segment_59,[must be inside]], [
 segment_57,[must be inside]], [segment_56,[right_of,under]], [segment_55,[
 must be inside]], [segment_49,[must be inside]], [segment_43,[above]], [segment_
 ,[right_of,above]]] are linked to segment_44

```

search for what
Enter one of segments_linked interpreted classify_segments location greater
less bye {bye}
interp
interpreted are [[segment_07,field],[segment_09,field],[segment_14,
field],[segment_18,field],[segment_26,field],[segment_40,not a field],[
segment_42,field],[segment_45,field],[segment_56,field],[segment_94,field],[
segment_95,field],[segment_96,field],[segment_98,field],[segment_99,field],[
segment_120,field],[segment_125,field],[segment_143,field],[segment_146,field],
[segment_151,field],[segment_163,field],[segment_176,field],[segment_181,field]
,[segment_199,field],[segment_212,field],[segment_240,not a field],[segment_242
,not a field],[segment_243,not a field],[segment_244,not a field]]
search for what
Enter one of segments_linked interpreted classify_segments location greater
less bye {bye}
bye

```

```

*** Search features environment ***
Enter one of find help bye {find}
bye

```

```

OK what do I do
Enter one of interpret merge both consult bye {bye}
interpret

```

```

segment_44 interpreted as field
Interpretation is by_evidence

```

```

some statistics :[segment_44,[34,lines_in_window],[15,lines_full]]

```

```

Processing segment_44

```

```

OK what do I do
Enter one of interpret merge both consult bye {bye}
bye

```

```

Which segment to focus on .

```

```

Enter one of bye segment_02 segment_07 segment_08 segment_09 segment_10
segment_13 segment_14 segment_18 segment_26 segment_40 segment_42 segment_44
segment_45 segment_47 segment_56 segment_94 segment_95 segment_96 segment_98
segment_99 segment_120 segment_125 segment_143 segment_146 segment_151
segment_163 segment_176 segment_181 segment_199 segment_210 segment_212
segment_240 segment_242 segment_243 segment_244 {bye}
bye

```

```

What do you want to do ?

```

```

Enter one of focus_all_segments focus_big_segments analyze_all_segments
quit {quit}
focus_all

```

```

Which segment to focus on
{bye}
segment_89

```

```

Processing segment_89

```

```
OK what do I do
Enter one of interpret merge both consult bye {bye}
both

impossible merge between segment_89 and segment_44
segment_89 interpreted as road
Interpretation is bottom_up

some statistics :[segment_89,[2,lines_in_window],[2,lines_full]]

Processing segment_89

OK what do I do
Enter one of interpret merge both consult bye {bye}
bye

Which segment to focus on
{bye}
bye

What do you want to do ?
Enter one of focus_all_segments focus_big_segments analyze_all_segments
quit {quit}
quit

name of the output file to save
{projet}
projet
success

bye.
* MProlog Interpreter Terminated *
*** Prolog Engineering Interface (level 1) ***
Enter one of load save add replace delete display interpreter bye help
{interpreter}
bye
*** User Interface ***
Enter one of run resume save quit exit help {help}
quit
Are you sure?
Enter one of yes no {yes}
yes
```